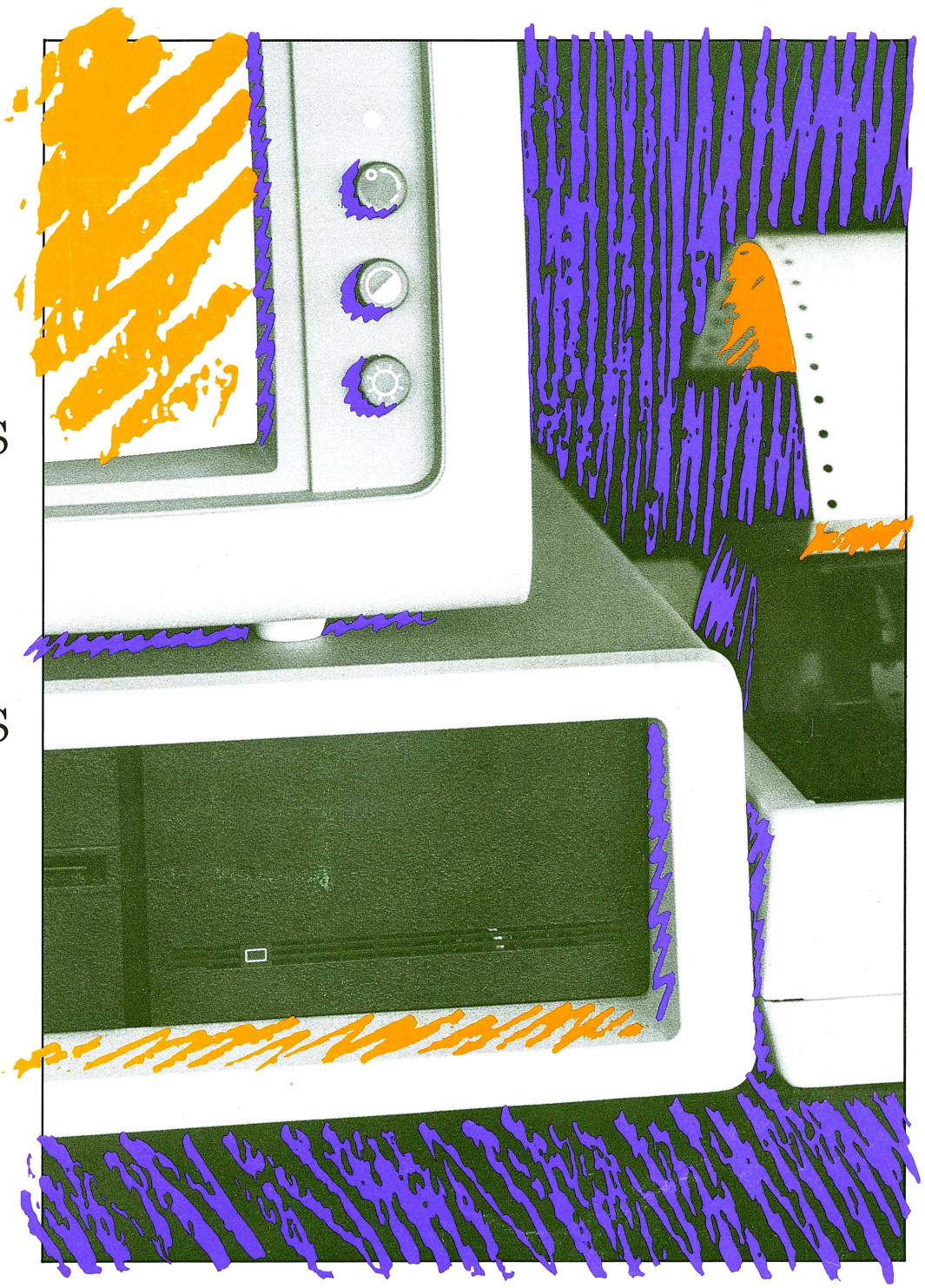
AN ASHTON-TATE SOFTWARE SUPPORT PUBLICATION

AUGUST 1986

AUG 21 1986

TechNotes

- Printing in Framework II
  - —Printer Drivers
  - —Common Problems
- dBASE III PLUS Change Summary
  - —Part Two
- dBASE III PLUS Help System
- Protect Notes



ASHTON-TATE™

TechNotes

TABLE OF CONTENTS

FEATURES

3 Framework II Printer Drivers
by Karen Robinson

10 Common Printer Problems and Their Solultions
by Chris Lo Vasco

12 dBASE III PLUS Change Summary—Part Two
by Joseph Perret

19 The Help System Revealed
by Karen Robinson

23 Protect Notes
by Richard Bunter and Karen Robinson

25 Automatic Record Locking While EDITing
by Doug Haman

27 Multiple Across Reporting Using LABEL FORM
by Karen Robinson

30 Searching for the First Date Within a Range
by Richard Goepel and Karen Robinson

35 Importing Delimited Files with No Delimiters in dBASE III
by John Thomas Regen

38 Advanced Sorting Primitives for dBASE TOOLS FOR C
by Robert Boies

54 Corrigenda

DEPARTMENTS

1 Read Me First
by Laurie Miller

48 Usage Tips

55 On-Line Services

59 MultiMate Technical Support

63 International Support

65 Letters

72 Index

READ ME FIRST

by Laurie Miller

In this month's issue of ***TechNotes***, we are featuring printing in Framework II. The first article on printing, "Common Printer Problems," by Chris Lo Vasco, outlines in a "Problem/Solution" format, several problems that Framework II users encounter most during printing. The second article on printing, "Framework II Printer Drivers," by Karen Robinson, presents complete information on printer drivers that cannot be found on disk or in the manual. Comprehensive and concise, the article features an informative table which illustrates supported printers, drivers that are used for these printers, and attributes supported by these drivers.

In keeping with our series on reporting in dBASE III PLUS, we offer you "Multiple Across Reporting Using LABEL FORM," by Karen Robinson. This presents a program which generates reports with record information in columns.

Other informative dBASE III PLUS articles: "The Help System Revealed," by Karen Robinson, is a must-read for the novice user. This provides a map of the dBASE III PLUS Help system with an overview of what each section covers.

For the more experienced user who is already familiar with the dBASE III PLUS password utility, Protect, we present "Protect Notes," by Richard Bunter and Karen Robinson. This is a series of tips on using Protect.

Also, you will find part two of the dBASE III PLUS Change Summary. This reports on anomalies that were published in the November 1985 through January 1986 issues of ***TechNotes***, which were fixed with the release of dBASE III PLUS. (Last month the "dBASE III PLUS Change Summary—Part One" reported on fixed anomalies reported in the February through October 1985 issues.)

International Support: This month's international technical material comes from Ashton-Tate's Mexico and Asian offices. The information from Mexico deals with assigning a foreign character set (in this case, Spanish) to the keyboard characters in Framework II. Asia reports on which versions of Ashton-Tate products are available in Japanese and which computers are supported by these versions.

MultiMate Technical Support: This month we have "Common Printer Questions" and "MultiMate Magic," presented in "What if" scenarios.

Special Acknowledgment: The ***TechNotes*** Staff would like to extend a special "thank you" to Samuel Feldman from the Forefront Development Center for generously sharing his time and knowledge with Karen Robinson for her article "Framework II Screen Drivers," which appeared in ***TechNotes*** last month.

Until next month, *Bon Appetit*!

TechNotes

Publisher
Ashton-Tate Software Support Center
Jay Hanson, Director
Roy Moore, Manager

Editor-in-Chief
Laurie Miller

Editors
Kent Irwin, Christopher White

Technical Writer
Karen Robinson

Assistant Editor
Olivier Biggerstaff

Technical Staff
Larry Abame, Stanley Ballenger, Rodney Barbati, Jeffrey Bersin, Robert Boies, Richard Bunter, Mike Carlisle, Ray Cortenbach, Fred Crowmover, Michael Fiore, Ken Getz, Richard Goepel, Peter Gold, Stephen Goodman, Mina Gussin, Doug Haman, Mike Hedblom, Clint Helton, Jack Hirsch, David Howlett, Errol Jackson, Matt Johnson, Rosaline Keenan, Gail Kulikowsky, Steven Kurasch, Jery Levy, Jorge Lizalde, Chris Lo Vasco, Tom Mangelsdorf, Jerry Mendes, Vincent Mendillo, Elizabeth Miranda, Jeffrey Moore, Kieth Mund, Joseph Perret, Jaspal Rattan, John Regen, Deborah Ridolfo, Steve Silverwood, Joseph Stoltz, Ralph Wakefield, Lisa Wanek, Quinn Wildman, Roger Wegehoft, Kari Wolff, Leonard Zerman

Contributing Editor (Multimate)
Liz Burke

Index
Olivier Biggerstaff

Administrative Staff
Kathy Smith

Illustrator
Stephen Goodman

Artistic Staff
D. A. Gray Bare, Tim Dunne, Jeff Elliott, Darrell Evers, Louis Hirsch, R. James Landers, Renée Maynard, Patty Pesek,

Product Development
Forefront Development Center:
Marilyn McMahon, Greg Stikeleather
Glendale Development Center:
Debby Moody

DISCLAIMER NOTICE

The information contained in this publication is designed to help you develop and improve your programming skills and to assist you in using Ashton-Tate products. Ashton-Tate assumes no responsibility whatsoever for the uses made of any of the software code in this issue, whether modified or unmodified.

If you discover what you believe is an anomaly (as defined under **Anomalies**) not presented in this issue or others, please send a detailed description of the possible anomaly to:

Ashton-Tate
Software Support Center
20101 Hamilton Ave.
Torrance, CA 90502-1319

If the reported anomaly can be reproduced by our technicians, it may be added to a future issue of ***TechNotes***.

The policies that are described in ***TechNotes*** apply only to the United States (and possessions), Canada, and Mexico. Residents of other foreign countries should contact a local Ashton-Tate office for policy information and technical support.

Ashton-Tate and the authors will not be liable for special, incidental, consequential, indirect or other similar damages, even if we have been advised of the possibility of such damages. This means we are not responsible for damages or costs incurred as a result of loss of time, loss of data, loss of profits or revenues, or loss of use of the software. In addition, we are not responsible for damages or costs incurred in connection with obtaining substitute software, claims by others, inconvenience, or similar costs.

No warranties, either expressed or implied, are made regarding the contents of this publication, its merchantability or fitness for any particular purpose, all of which are hereby expressly disclaimed. Note, however, that some states do not allow the exclusion or limitation of warranties or of direct, incidental or consequential damages, so these warranty limitations may not apply to you.

In no event will our liability for any damages to you or any other person ever exceed the price paid for your subscription to ***TechNotes***, regardless of any form of the claim.

Additional statements by agents, employees, distributors and dealers of Ashton-Tate do not constitute warranties by Ashton-Tate and do not bind Ashton-Tate.

TechNotes (ISSN 0883-6493) is an Ashton-Tate Software Support Center publication containing technical reference information about three of Ashton-Tate's products: dBASE II®, dBASE III PLUS®, and Framework II™.

Entire contents Copyright © Ashton-Tate 1986. All rights reserved. Commercial reproduction of the material appearing in ***TechNotes*** is forbidden without prior written permission. Direct all requests for reproduction permission to Ashton-Tate Software Support Center, 20101 Hamilton Ave., Torrance, CA 90502-1319 ATTN: Laurie Miller.

TechNotes is published monthly at a subscription price of $50.00 for one year. Additional rates: Canada and Mexico, $65.00; other countries, $75.00. Monthly issues are available in back issues for $10.00 each (1985 issues) and $7.00 each (1986 issues). ***TechNotes*** is published by the Ashton-Tate Software Support Center, 20101 Hamilton Ave., Torrance, CA 90502-1319. POSTMASTER: Send subscription inquiries and address changes to ***TechNotes*** Subscription Service, P.O. Box 3729, Escondido, CA 92025-0929.

dBASE II and dBASE III are registered trademarks of Ashton-Tate. dBASE, RunTime, FRED, Framework, and ASHTON-TATE are trademarks of Ashton-Tate. Some of the trademarks of other companies mentioned in this issue appear for identification purposes only.

The Ashton-Tate Software Support Center's hours are 6:00 a.m. to 4:30 p.m., PST, Monday through Friday, except on Thursday when we close at 4:00 p.m. To reach our technicians for technical support, call (213) 329-0086. Be sure to have your software serial number ready for the technician.

Framework II Printer Drivers

by Karen Robinson

This article is a comprehensive discussion of Framework II printer drivers. Included are sections on the supported features for each printer, printing graphs, and sending special control codes to your printer. There is also a discussion of the new printer drivers that are available, and instructions on how to install them. At the end of the article is a table that lists all of the supported printers and their respective attributes.

Printing Attributes

The printing attributes that Framework II offers are bold, underline, italics, condensed, and correspondence-quality print. Whether or not Framework II supports these features depends on the capability of your printer. If your printer supports them, so does Framework II. If your printer does not support a feature, Framework II uses an alternate print style. The alternate for bold is doublestrike, for underline and italics, underscore. If graphics are not supported by a driver, then graphs are printed using text characters instead of dot graphics. If correspondence-quality print is not supported, then Framework II simply ignores the request.

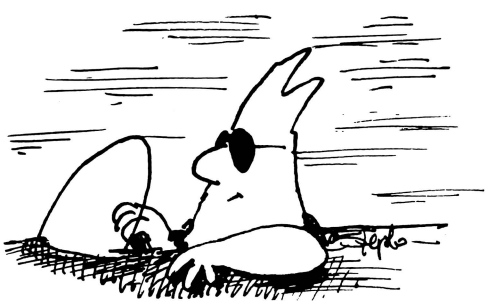


Table 1, at the end of this article, lists the supported printers and print attributes available through Framework II. If the table indicates "Yes" for an attribute, Framework II sends out the appropriate codes for that particular printer; otherwise, the alternate print style is used. If your printer is not listed, use the generic driver. Graphs will print with text characters, bold will print as double strike, and underline and italics as underscores. However, if your printer supports any of these attributes, you can embed the control codes within the document (more on this later). If your printer emulates another, you may be able to use that driver; however, graphs probably won't print correctly.

Printing Graphs

For some printers, Framework II offers a choice of graphic densities: low, standard, and high resolution. The lower the resolution of the printer driver, the larger your printed graphs can be. So, if you choose a high resolution driver, expect your printed graphs to be small.

Sending Printer Codes

As stated earlier, in addition to the print features supported through the Framework II menus, you can also send your own set of printer codes, allowing you to use any print style supported by your printer. These codes can be sent from the frame level, which will affect the contents of the entire document, or from within the document, which can affect any portion of the document.

From the frames level, printer control codes can be sent with the **Starting Control Codes** option or with the **@st** function. Within the **Starting Control Codes** menu option, each ASCII code is entered as a three-digit number, delimited with brackets { }. For example, on an Epson FX-80 printer, the ASCII codes for expanded-compressed print are 20 and 18. The correct format for these codes is {020}{018}. On the other hand, the **@st** function is entered in the formula area of the frame. Using the same example, the correct format is **@st**(''{020}{018}''). In order for the **@st** formula to be recognized, **Recalc Formula during Printing** must be set ON (found in the Print:Control Options menu).

From within a document, you can embed printer codes within the text using the **Alt** key and the numeric keypad. This allows you to affect the print style of any portion of the document. The procedure is as follows:

1. Move the cursor to the desired location in the word frame.
2. Press and hold down the **Alt** key while entering the digits of the first ASCII code on the numeric keypad. Then release the **Alt** key, and a special character will appear on the screen.
3. Repeat step 2 for each ASCII code in the sequence. Using the example above, the correct procedure is **Alt-20**, **Alt-18**.
4. To turn off a print attribute, move the cursor to the appropriate place in the word frame and then repeat steps 2 and 3, substituting the codes which will turn off the attribute.

The international character set, superscripts, and subscripts are also sent using this procedure. Naturally, the success of such print styles is dependent on whether your printer supports these features. The chart below shows the ASCII codes that are used by Framework II for superscripts and subscripts, regardless of the printer used.

ASCII Decimal Code	Mode
244	Superscript ON
245	Superscript OFF
246	Subscript ON
247	Subscript OFF

The problem with using special printer codes, including proportional spacing and microjustification (which are not supported in Framework II), is that they may throw off the line spacing when the document is printed. This happens because Framework II treats the codes as ordinary characters. If their presence within your document affects line or paragraph formatting, you will have to manually adjust your formatting by adding or subtracting spaces.

Special Considerations

IBM QuietWriter Model 2

If you have an IBM QuietWriter Model 2 printer, install it as an IBM Graphics Printer (IBMGP.PR). All of the attributes supported on the IBM Graphics Printer are also supported on the IBM QuietWriter Model 2. Do not install it as a QuietWriter Model 1; the two models use very different code sets.

Epson LQ1500

On an Epson LQ1500, graphs always print in high resolution; hence, the largest possible graphs are fairly small. Attempting to size the graph larger may return the error message "Frame too large to draw graph." The Ashton-Tate Software Support Center and on-line support forums now have a new driver for the Epson LQ1500 that will print graphs in standard resolution.

New Drivers Available

There are four new drivers available from the Software Support Center or the Ashton-Tate forum on CompuServe. These new drivers support the following printers:

Tandy CGP220
NEC Pinwriter P5
Okidata Microline 84
Epson LQ1500 (for standard resolution graphs)
[Others may be available by the time you read this]

To install any of the new drivers not included as part of the Setup program, follow these steps:

1. From the DOS prompt, type Setup and then press **Return**.
2. At the first SETUP screen, select (2) ALL OTHER USES OF THE SETUP PROGRAM.
3. Next, select the drive and directory where Framework II is installed. This is usually (2), the current default drive and directory.
4. Select (2) CONFIGURATION.
5. Select (1) PRIMARY HARDWARE.
6. Select (3) PRINTER 1 DRIVER.
7. Select (8) I WANT TO ENTER MY OWN DRIVER FILENAME.
8. Enter the driver filename and then press **Return**. Be sure to include the file extension.
9. Press M for MAIN MENU.
10. Select (7) SAVE ALL NEW SETTINGS.
11. When the prompt "Please insert your setup disk in drive A:" is displayed, place the disk that contains the new driver file into drive A:. Then follow the prompts on the screen to finish saving the changes.

Supported Printers

Table 1 lists the supported printers, the drivers that are used for these printers, and the supported attributes. For more information on printers and printing see "Common Printer Problems and Their Solutions" in this issue.

Table 1. Framework II Printer Drivers

Manufacturer	Model	Driver Filename	Bold	Italic	Underline	Graphs	Compressed	Letter Quality
Anadex	9625B (Silent Scribe)	ANA9625B.PR	Yes	Underscore	Underscore	Yes	Yes	Yes
Brother	HR-15	HR15.PR	Yes	Underline	Yes	Yes	Yes	N/A
	HR-25	HR15.PR	Yes	Underline	Yes	Yes	Yes	N/A
C-Itoh	1550A, B, BPI	CI1550A.PR	Yes	Underline	Yes	Yes	Yes	No
	1550S	CI1550S.PR	Yes	Yes	Yes	Yes	Yes	No
	1550SC[a]	CI1550SC.PR	Yes	Yes	Yes	Yes	Yes	No
	8510A, B, BPI	CI8510A.PR	Yes	Underline	Yes	Yes	Yes	No
	8510S	CI8510S.PR	Yes	Yes	Yes	Yes	Yes	No
	8510SC[a]	CI8510SC.PR	Yes	Yes	Yes	Yes	Yes	No
	3500 model 20	CI3520.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Prowriter Jr.							
	Standard graphs	DP_8050.PR	Yes	Underline	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
Data South	220	DS220.PR	Double	Underscore	Underscore	Yes	Yes	Yes
DataProducts	P-80	DP_P80.PR	Double	Underscore	Underscore	Yes	Yes	Yes
	P-132	DP_P80.PR	Double	Underscore	Underscore	Yes	Yes	Yes
	SPG-8050, SPG-8070 (no color)	DP_8050.PR	Yes	Underline	Yes	Yes	Yes	Yes
	SPG-8050, SPG-8070 (4 color)[a]	DP_8050C.PR	Yes	Underline	Yes	Yes	Yes	Yes
Diablo	(Xerox) 630 API	DIAB630A.PR	Yes	Underline	Yes	Yes	No	N/A
Epson	MX-80	GENERIC.PR	Double	Underscore	Underscore	No	No	No
	MX-80 with Graftrax	MX80G.PR	Yes	Yes	Yes	Yes	Yes	Yes
	MX-80 with Graftrax PLUS							
	Standard graphs	FX80_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	MX-100	GENERIC.PR	Double	Underline	Yes	No	No	No
	MX-100 with Graftrax	MX80G.PR	Yes	Yes	Yes	Yes	Yes	Yes
	MX-100 with Graftrax PLUS							
	Standard graphs	FX80_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	RX-80, FX-80, FX-80 PLUS							
	Low resolution graphs	FX80_1.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Standard graphs	FX80_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	RX-80/FX-80 with 8177	EPSN8177.PR	Double	Yes	Yes	Yes	Yes	Yes
	RX-100, FX-100, FX-100 PLUS							
	Standard graphs	FX80_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	RX-100/FX-100 with 8177	EPSN8177.PR	Double	Yes	Yes	Yes	Yes	Yes
	FX-85							
	Low resolution graphs	FX85_1.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Standard graphs	FX85_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX85_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	FX-185							
	Low resolution graphs	FX85_1.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Standard graphs	FX85_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX85_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	DX-10	DIAB630A.PR	Yes	Underline	Yes	Yes	No	N/A
	DX-20	DIAB630A.PR	Yes	Underline	Yes	Yes	No	N/A
	DX-35	DIAB630A.PR	Yes	Underline	Yes	Yes	No	N/A
	JX-80 with 8177	EPSN8177.PR	Double	Yes	Yes	Yes	Yes	Yes
	JX-80 (8 color) [a]	JX80.PR	Yes	Yes	Yes	Yes	Yes	Yes
	LQ-1500							
	Standard graphs[b]	LQ1500_2.PR						
	High resolution graphs	LQ1500.PR	Yes	Yes	Yes	Yes	Yes	Yes

Footnotes are listed at the end of this table.

Table 1. Framework II Printer Drivers (cont.)

Manufacturer	Model	Driver Filename	Bold	Italic	Underline	Graphs	Compressed	Letter Quality
	SQ-2000	LQ1500.PR	Yes	Yes	Yes	Yes	Yes	Yes
	LX-80	LX80.PR	Yes	Yes	Yes	Yes	Yes	Yes
	LX-90	LX80.PR	Yes	Yes	Yes	Yes	Yes	Yes
Facit	4512	FAC4512.PR	Yes	Underline	Yes	Yes	Yes	Yes
Fujitsu	DotMax 9f							
	Standard graphs	FX80_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	High resolution graphs	FX80_3.PR	Yes	Yes	Yes	Yes	Yes	Yes
	DotMax 9i	IBMGP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	DotMax 24i	FUJ24I.PR	Yes	Yes	Yes	Yes	Yes	Yes
	DotMAx 24c (8 color) [a]	FUJ24C.PR	Yes	Yes	Yes	Yes	Yes	Yes
Generic Driver	Any printer not listed	GENERIC.PR	Double	Underscore	Underscore	No	No	No
HP	ThinkJet	HPJET.PR	Yes	Underline	Yes	Yes	Yes	No
	LaserJet							
	In 100 dpi graphics mode	HPLAS100.PR	Double	Yes	Yes	Yes	No	No
	In 150 dpi graphics mode	HPLAS150.PR	Double	Yes	Yes	Yes	No	No
	In 300 dpi graphics mode	HPLAS300.PR	Double	Yes	Yes	Yes	No	No
	Landscape	HPLASL.PR	Double	Underline	Yes	Yes	Yes	No
IBM	Graphics Printer	IBMGP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	ProPrinter							
	Low resolution graphs	IBMPRO_1.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Standard graphs	IBMPRO_2.PR	Yes	Underline	Yes	Yes	Yes	Yes
	High resolution graphs	IBMPRO_3.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Color JetPrinter (Monochrome)	IBMCJET2.PR	Yes	Underline	Yes	Yes	Yes	Yes
	QuietWriter (model 1)	IBMQUIET.PR	Double	Underline	Yes	Yes	No	No
	WheelPrinter	IBMWHEEL.PR	Double	Underline	Yes	Yes	Yes	N/A
NEC	Pinwriter P2	IBMGP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Pinwriter P3	IBMGP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Pinwriter P5[b]	NECP5.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Spinwriter 3550	NEC3550.PR	Double	Underline	Yes	Yes	Yes	N/A
	Spinwriter 7710	NEC7710.PR	Yes	Underline	Yes	Yes	No	N/A
	Spinwriter 8850	NEC8850.PR	Yes	Underline	Yes	Yes	Yes	N/A
OKIDATA	Microline 80	GENERIC.PR	Double	Underline	Yes	No	No	No
	Microline 82	GENERIC.PR	Double	Underline	Yes	No	No	No
	Microline 82A with Okigraph	OKI82AGR.PR	Double	Underline	Yes	Yes	Yes	No
	Microline 83A	GENERIC.PR	Double	Underline	Yes	No	No	No
	Microline 83A with Okigraph	OKI82AGR.PR	Double	Underline	Yes	Yes	Yes	No
	Microline 84[b]	OKI84.PR	Double	Underline	Yes	Yes	Yes	Yes
	Microline 182	OKI182.PR	Double	Underline	Yes	Yes	Yes	Yes
	182 Plug 'n Play	OKI182PN.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Microline 183	OKI182.PR	Double	Underline	Yes	Yes	Yes	Yes
	183 Plug 'n Play	OKI182PN.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Microline 92	OKI92_2.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Microline 93	OKI92_2.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Microline 192	OKI192_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Microline 193	OKI192_2.PR	Yes	Yes	Yes	Yes	Yes	Yes
	92 or 192 with Plug 'n Play	OKI92PNP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	93 or 193 with Plug 'n Play	OKI92PNP.PR	Yes	Underline	Yes	Yes	Yes	Yes
	Pacemark 2350	OKI2410.PR	Double	Underline	Yes	Yes	Yes	Yes
	Pacemark 2410	OKI2410.PR	Double	Underline	Yes	Yes	Yes	Yes
	Okimate-20	IBMGP.PR	Yes	Underline	Yes	Yes	Yes	Yes
Qume	Sprint 11 Plus	QUME11.PR	Yes	Underline	Yes	Yes	Yes	N/A
	LetterPro 20	QUME11.PR	Yes	Underline	Yes	Yes	Yes	N/A

Footnotes are listed at the end of this table.

Table 1. Framework II Printer Drivers (cont.)

Manufacturer	Model	Driver Filename	Bold	Italic	Underline	Graphs	Compressed	Letter Quality
Star Micronics	Gemini-10x	GEM10.PR	Yes	Yes	Yes	Yes	Yes	Yes
	Gemini-15x	GEM10.PR	Yes	Yes	Yes	Yes	Yes	Yes
TI	855	TI855.PR	Yes	Yes	Yes	Yes	Yes	Yes
	865	TI855.PR	Yes	Yes	Yes	Yes	Yes	Yes
Tandy	CGP220[b]	CPG220.PR	Yes	Yes	Yes	Yes	No	No
Toshiba	P1340	P1340.PR	Double	Underline	Yes	Yes	Yes	Yes
	P1351	P1340.PR	Double	Underline	Yes	Yes	Yes	Yes
	P341	P351.PR	Yes	Yes	Yes	Yes	Yes	Yes
	P351	P351.PR	Yes	Yes	Yes	Yes	Yes	Yes
Xerox	(Diablo) 630 API	DIAB630A.PR	Yes	Underline	Yes	Yes	No	No

[a]This driver prints multicolored graphs.
[b]This driver is available from the Ashton-Tate Software Support Center or the Ashton-Tate Forum on CompuServe.

Common Printer Problems and Their Solutions

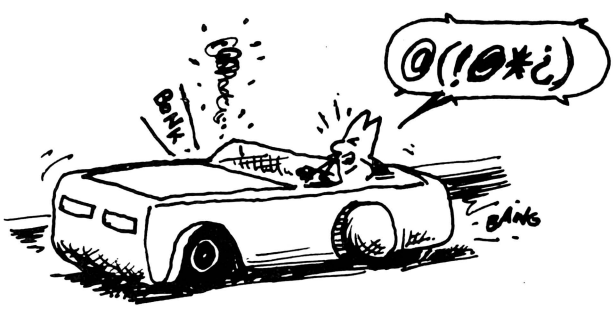
by Chris Lo Vasco

Some users have expressed confusion regarding the interaction of printers, printer drivers, and the Print menu options. This article addresses the four most common printer problems and their most likely solutions.

Problem: Misaligned printout after the first page. That is, the first page seems to print just fine, but the second and successive pages are misaligned vertically.

Solution: This is almost always caused by not properly setting the top of form on the printer. One of the ways to insure that the top of form is set properly is to set the horizontal perforation of the paper at the print head and then turn the printer off and on. This step is critical because Framework II does not formfeed at the top of the first page. Framework II starts printing, and counting lines, from the current position of the print head, regardless of the top of form setting on the printer. Top of form may not be set correctly if the printer was used outside of Framework II, and has not been reset since then. After the last line on a page has been printed, a formfeed is sent to the printer, which will cause the paper to drop down to the printer's top of form. The second and successive pages will be misaligned.

You can accidentally fall into this trap if you print something with another program, and then print a document using Framework II, without resetting the top of form. If, for instance, you are in a DOS access frame and TYPE a text file to the printer, the printer will not necessarily be left at the top of form. You might easily exit the DOS Access frame, and later decide to print a frame. If, at this point, you manually align the printer carriage to the top of a page by turning the knob on the side of the printer, you will throw off the top of form. When manually aligning printer paper, you should always turn the printer off and on before beginning to print.



Problem: Successive pages print lower and lower on the physical page (page creep).

Solution: The key issue for this problem is form length. Generally, you will have page creep if the printer form length or Framework II's **Lines per Page** is set to inappropriate values for the physical page. The solution is to properly set up the printer for the physical page size you are going to print. You can refer to the printer's manual to find out how to do this on your printer. For most applications, **Lines per Page** should also be set to the number of lines that will physically fit on a page (usually 66 lines, for an 8 1/2'' x 11'' page).

Problem: Printout does not match document on screen.

Solution: Printed documents which page-break differently than on the screen may be caused by the header and footer settings. **Top Margin**, **Down from Page Top**, and the number of lines in the actual header must all add up. For example, trying to print a four-line header with the **Top Margin** equal to six and the **Down from Page Top** equal to four will result in a printed document that is paginated differently from the screen display with **View Pagination** ON. Headers and footers must fit entirely within the top or bottom margins, and must be preceded and followed by at least one blank line. The same precautions that apply to headers naturally apply to footers as well. For a good visual reminder of Framework II page layout, refer to Figure 7-4 on page 7-12 of *Using Framework II*.

Problem: Garbage characters appear on printout, or some printing features do not seem to work.

Solution: Very often this is caused by installing an improper printer driver. For example, refer to the Okidata series of printers. An important option for using the Okidatas is the precise physical configuration of the printer. For example, the Microline 92 driver will not work properly when installed as a Microline 92 with Plug'n'Play. The reason for this is that the Microline 92 with Plug'n'Play only understands IBM printer codes and the Microline 92 only understands Okidata printer codes. This incompatibility can usually be diagnosed by the appearance of a three-digit number appearing at the top left of the printout. This number is the **Offset from Left** control code, and it is printing because it is an improper printer control code for that configuration. If you have an unsupported printer and are having some difficulty with the printing features, use Setup to install the Generic driver. The Generic printer driver is the only driver that is designed to work with equipment that is unknown to Framework II.

If you're experiencing printer problems, please run through this entire list. Your problem may be caused by one or more of these errors.

dBASE III PLUS Change Summary—Part Two

by Joseph Perret

This is the second of a four-part series listing anomalies reported in ***TechNotes*** for dBASE III version 1.1 and the Developer's Release fixed in dBASE III PLUS. The set was originally reported in issues beginning with November 1985 and ending with January 1986. As was the case last month, each entry is followed by an explanation of the change made to the command or function. Some anomalies have been repaired outright and now work as documented. Others, occurring when an internal limit was breached or incorrect syntax employed, now produce appropriate error messages.

Next month, we will publish the third part of the change summary for ***TechNotes*** issues February 1985 through April 1985.

BROWSE

Formerly, if you BROWSEd an indexed file, the record pointer was thrown off upon exiting. A SKIP command would act unpredictably and a command with a NEXT clause would include the current record twice or only the current record.

In dBASE III PLUS, BROWSE does not affect the record pointer and SKIP and NEXT work properly after BROWSE is exited.

Reported in the January 1986 issue of ***TechNotes***, page 45.

CTOD( )

Formerly, the CTOD( ) function accepted any string as an argument and returned a date. If the string was not a valid date string within the format of SET DATE TO, the results of the CTOD( ) function would be unusual.

The CTOD( ) function converted any literal that contains only non-numeric characters to 11/30/99. For example:

```
? CTOD("HALLOWEEN")
11/30/99
```

The CTOD( ) function converted strings containing numeric and non-numeric characters not in the format selected with SET DATE TO to a date without producing an error message. For example:

```
? CTOD("12//12")
11/30/12
? CTOD("he12/12")
11/30/11
? CTOD("12/12/he")
12/12/00
```

Now, the Developer's Release and dBASE III PLUS return a blank date if the CTOD( ) function is used with a literal in this way.

Reported in the November 1985 issue of ***TechNotes***, page 28.

DO

Formerly, executing a command file with the same name as a procedure in a procedure file previously opened and closed, caused the return pointers to be lost. The following program demonstrates a circumstance where the problem occurs.

```
* —-Test.PRG
DO WHILE .T.
    SET PROCEDURE TO Testa
    DO Test1 <— From Testa.PRG.
    CLOSE PROCEDURE
    DO Test1 <— Separate command file
ENDDO <— will terminate although no
              EXIT or RETURN has been executed.
```

This problem is corrected in the Developer's Release and dBASE III PLUS.

Reported in the December 1985 issue of ***TechNotes***, page 32.

GOTO

Formerly, attempting to GOTO a very large number (approximately 44 digits), with a file in USE might have locked the computer, requiring a warm or cold boot.

Now, the Developer's Release and dBASE III PLUS properly return the error message "Record is out of range" when this is done.

Reported in the January 1986 issue of ***TechNotes***, page 46.

HISTORY (Developer's Release)

Formerly, the maximum command line length was 254 characters. Since HISTORY reserves 240 characters (three lines of the screen) for each command line, commands longer than 240 characters were truncated when recorded in HISTORY. This occurred whether the command was recorded through SET HISTORY ON or SET DOHISTORY ON.

In dBASE III PLUS, the three-line format for HISTORY is no longer used. In its place, the command line is a 254-character scrollable edit space allowing commands to be entered up to the 254-character limit.

Reported in the November 1985 issue of ***TechNotes***, page 28.

IIF (Developer's Release)

In the Developer's Release, nesting the IIF( ) function eight or more levels deep returned a zero instead of the designated value.

Now, dBASE III PLUS properly executes as many nested IIF( ) statements as can fit within the 254-character command line limit.

Reported in the December 1985 issue of ***TechNotes***, page 34.

INDEX

Formerly, you could not build an index file using the SUBSTR( ) function if the second parameter of the SUBSTR( ) function contained the LEN( ) function. For example:

INDEX ON SUBSTR(<Field name>,1,LEN(<Field name>)) TO Indexfile

When built, the index file appeared to be correctly constructed. It was then possible to FIND or SEEK a record and EDIT it. However, the index file was not updated and any attempt to GO TO a record returned the error message "Record not in index."

This is corrected in the Developer's Release and dBASE III PLUS.

Reported in the November 1985 issue of ***TechNotes***, page 32.

LOAD (Developer's Release)

Formerly, a file with an eight-character filename would not be LOADed correctly. For example:

LOAD Diskread
CALL Diskread

would return:

File was not LOADed.

when the CALL command was executed. The LOAD command produced no error message. If the filename was changed to Dskread.BIN, the file LOADed properly.

Now, dBASE III PLUS properly LOADs files having names of eight characters without error.

Reported in the November 1985 issue of ***TechNotes***, page 32.

MODIFY COMMAND

Formerly, when using **Ctrl-K-R** in the dBASE III word processor to read an external text file, the length of the file name including drive and path was limited to 15 characters. If its length was more than 15, the computer might freeze with the message "File does not exist (Hit SPACE)" or the error message "*** STACK OVERFLOW ***" would appear and the user would return to the operating system.

In the Developer's Release and dBASE III PLUS, the read file function supports filenames, including paths, longer than 15 characters.

Reported in the December 1985 issue of ***TechNotes***, page 32.

MODIFY COMMAND (Developer's Release)

Formerly, when MODIFY COMMAND was used to create a file, pressing **Ctrl-@**, **Ctrl-C**, or **PgDn** on the first line of text would frequently cause graphic characters to appear. These could be deleted.

Now in dBASE III PLUS this no longer occurs.

Reported in the November 1985 issue of ***TechNotes***, page 33.

MODIFY REPORT

Formerly, there were problems with the column width setting. If a change in the field contents affected the column width, the width setting changed automatically to reflect the new field content width. If however, the new value was not explicitly re-entered in the setting area, the width reverted to the original value when another column was accessed or CREATE/MODIFY REPORT was exited.

Now, in dBASE III PLUS, this no longer occurs. Changes made to the field contents are reflected in the column width and need not be entered explicitly.

In the Developer's Release, some modifications were made to rectify this problem. When the field contents changed, the column width reflected this change only if the column width was increased. However, unlike versions 1.0 and 1.1, all changes to the column width were saved when CREATE/MODIFY REPORT was exited.

In dBASE III PLUS, changes to the field contents are reflected in the width setting whether or not the change increases or decreases the size of the column width.

A related problem occurred when the column width was less than the length of the field contents expression. When you MODIFYed REPORT, the width of the column reverted to the length of the expression forcing you to explicitly reset it every time the REPORT was MODIFYed. This anomaly was not fixed in the Developer's Release.

In dBASE III PLUS the original value of the column width is kept until the field contents are changed. At that point the column width changes to reflect the new width.

Reported in the November 1985 issue of ***TechNotes***, page 33.

PARAMETERS

Formerly, a PARAMETERS command line could not be wrapped to the next line with a semicolon. This would produce the error message, "Syntax error", when the semicolon was encountered. If, however, the command was allowed to wrap automatically in MODIFY COMMAND, it executed properly.

Now, the Developer's Release and dBASE III PLUS allow the PARAMETERS statement to be wrapped with a semicolon.

Reported in the January 1986 issue of ***TechNotes***, page 46.

RANGE

Formerly, it was possible to define a RANGE from the upper limit to the lower limit rather than from the lower to the upper. For example:

```
mem = 0
@ 2,2 GET mem PICTURE "9" RANGE 9,1
```

Although this was not trapped as an error, no entry could meet this RANGE specification. Placing the higher value as the first argument created an impossible condition by requiring that the entry be both greater than or equal to the upper limit and less than or equal to the lower limit.

Now, in dBASE III PLUS, such a RANGE statement returns the error message "Syntax error."

Reported in the December 1985 issue of ***TechNotes***, page 28.

READ

In the Developer's Release, it was not possible to issue a READ command from within the HISTORY mode. Any attempt to READ from HISTORY positioned the cursor at 0,39 on the screen. Pressing any key would return the user to the dot prompt.

Now in dBASE III PLUS, a READ entered in HISTORY executes like any other command.

Reported in the January 1986 issue of ***TechNotes***, page 40.

REPLACE

Formerly, using an incomplete REPLACE command (REPL, REPLA, REPLAC) or REPLACE without an argument could result in the error message "Illegal opcode <no.>." The opcode number was not significant and varied from machine to machine. The Developer's Release of dBASE III returned "1 Records replaced."

Now, dBASE III PLUS returns the error message, "Syntax error" if any of these above permutations of the REPLACE command are used.

Reported in the November 1985 issue of ***TechNotes***, page 33.

SET HISTORY TO (Developer's Release)

1. Formerly, there was no way to escape from LIST HISTORY. If a user SET HISTORY TO a large value, entered a large number of commands, and LISTed HISTORY, the computer might have been tied up for some time as all of recorded HISTORY scrolled by.

   Now in dBASE III PLUS, it is possible to escape from LIST HISTORY, if SET ESCAPE is ON.

   Reported in the November 1985 issue of ***TechNotes***, page 33.

2. Formerly, there was no error trapping on the upper limit of HISTORY, and it was possible to SET it to a value higher than 16,000. Values less than 1,000,000 seemed to work properly, although no test was performed to determine if more than 16,000 lines of code were stored.

   When the value of HISTORY exceeded 1,000,000, the HISTORY buffer became corrupted. Any subsequent attempt to DISPLAY or LIST HISTORY displayed graphic characters to the screen and might freeze the computer.

   Now, dBASE III PLUS returns the error message "Illegal value" if values greater than 16,000 are given to the SET HISTORY TO command.

   Reported in the November 1985 issue of ***TechNotes***, page 34.

SET ORDER TO (Developer's Release)

1. Formerly, SET ORDER TO any index file in USE would take an inordinate amount of time to return control to the program or dot prompt after GO BOTTOM. The result of SETting ORDER TO would take from six to ten times as long as SETting INDEX TO the same order and restoring the position of the record pointer to the bottom of the master index file.

   In dBASE III PLUS, there is no noticeable difference in speed between these two operations.

   Reported in the November 1985 issue of ***TechNotes***, page 29.

2. In the Developer's Release, if the order of open indexes was changed with SET ORDER TO, REINDEX would reset the order to its original state. For example:

   ```
   USE Test INDEX Test1, Test2    <-- Test1.NDX is master.
   SET ORDER TO 2                 <-- Test2.NDX is master.
   REINDEX                        <-- Test1.NDX is master.
   ```

   Now, dBASE III PLUS retains the master index chosen with SET ORDER TO after a REINDEX command is executed.

   Reported in the January 1986 issue of ***TechNotes***, page 40.

SET TALK ON

Formerly, for commands that produce an ODOMETER reading, such as INDEX or COUNT, SET TALK ON took precedence over SET CONSOLE OFF so that the results of TALK were displayed when SET CONSOLE was OFF.

Now in the Developer's Release and dBASE III PLUS, SET CONSOLE OFF suppresses TALK even if it is SET ON.

Reported in the December 1985 issue of ***TechNotes***, page 33.

SORT

Formerly, SORTing to a file with an extension other than .DBF would result in the extension being ignored and a .DBF extension given to the new file.

Now, dBASE III PLUS allows SORT to create files with extensions other than the default.

Reported in the December 1985 issue of ***TechNotes***, page 33.

TYPE

Formerly, use of the incorrect syntax,

TYPE <File> TO SCREEN

was trapped as an error, giving the error message "File does not exist." Subsequent attempts, however, to USE or MODIFY COMMAND that file returned the error message "File is already open." Neither CANCEL nor CLEAR ALL would close the file. Quitting and re-entering dBASE III was the only option.

Now, attempting to TYPE a file TO SCREEN in the Developer's Release and dBASE III PLUS returns the error message, "Syntax error" and the file being TYPEd remains properly closed.

Reported in the November 1985 issue of ***TechNotes***, page 35.

The Help System Revealed

by Karen Robinson

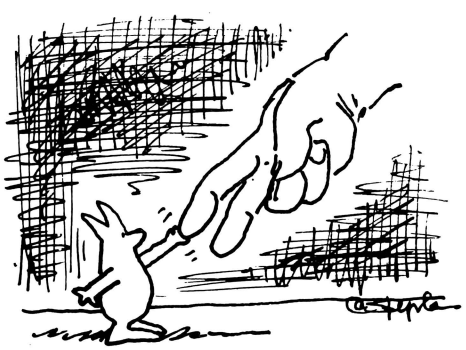
The dBASE III PLUS Help system is an excellent source of information for both new and seasoned users. It's also fast and easy to use. Information on every dBASE III PLUS command and function is available. There is also information on memory variables, field lists, and expressions including usage instructions. In addition, there is a task-oriented section that explains how to open, close, COPY, DELETE, or RENAME a file.

There are three ways to enter HELP.

1. Type HELP, or press **F1** at the dot prompt, and the main menu of HELP displays. From here, you can make choices to navigate to the desired information screen.
2. Type HELP <command>, where <command> is the dBASE III PLUS command or function that you want to know more about. You then directly access a screen that explains the syntax and usage of the dBASE III PLUS command.
3. HELP is also available when you have HELP SET ON and you get an error in a command executed at the dot prompt. When this happens, you are presented with the question "Do you want help (Y/N)?". If you answer in the affirmative, a HELP screen appears. Notice that HELP is context sensitive, but only to the command verb and not to other parts of the command line where errors often occur.

   If you find this automatic prompting for HELP bothersome, you can suppress it by SETting HELP OFF.

Each screen in HELP is structured similarly. The top right side of each screen displays the name of the screen. The middle portion contains either the menu items or the requested information. The bottom portion of the screen displays the keys used to navigate through HELP. At the bottom of the screen is an "ENTER>" prompt. At this prompt, you can enter a dBASE III PLUS command, a screen name (found at the upper right of the screen), or a menu item number, all which will navigate you to the appropriate information screen.



Navigation through HELP is quite easy. Use **Uparrow** and **Dnarrow** to position the cursor highlight to an option on a Menu screen or the "ENTER >"prompt. Press **Return** to select the highlighted option. At the "ENTER >" prompt, near the bottom of each HELP screen, you can enter a command, screen name (found at the upper right of the screen), or menu item number.

The table below summarizes the navigation keys and their functions.

Key Name	Action
Esc	Exits HELP to the dot prompt.
Arrow keys	Position the highlighted cursor to a menu option.
PgUp/PgDn	Moves to the next or previous screen.
F10	Displays the previous menu.

The main menu of HELP is made up of six main sections, each with a variable number of sub-levels. The following is a brief overview of each section.

Getting started is for the first-time user. It discusses how to invoke and move through HELP. It also defines the special symbols used in the syntax of dBASE III PLUS commands.

What is a... is a glossary of terms used within the syntax of dBASE III PLUS commands and functions. Some of the terms covered are: expressions and conditions, field lists, key fields, and memory variables. The HELP screens provide a definition of each item and an example of how to use it.

How do I... lists those tasks most important to new users, such as leaving dBASE III PLUS, opening and CLOSEing database files, and RENAMEing or DELETEing database files. The list doesn't stop here; refer to the HELP map below for a broader view.

Creating a database file gives a narrative description of a database file, rules for naming files and fields, and choosing field types and sizes. If you are confused about creating a database file, this is an important section for reference.

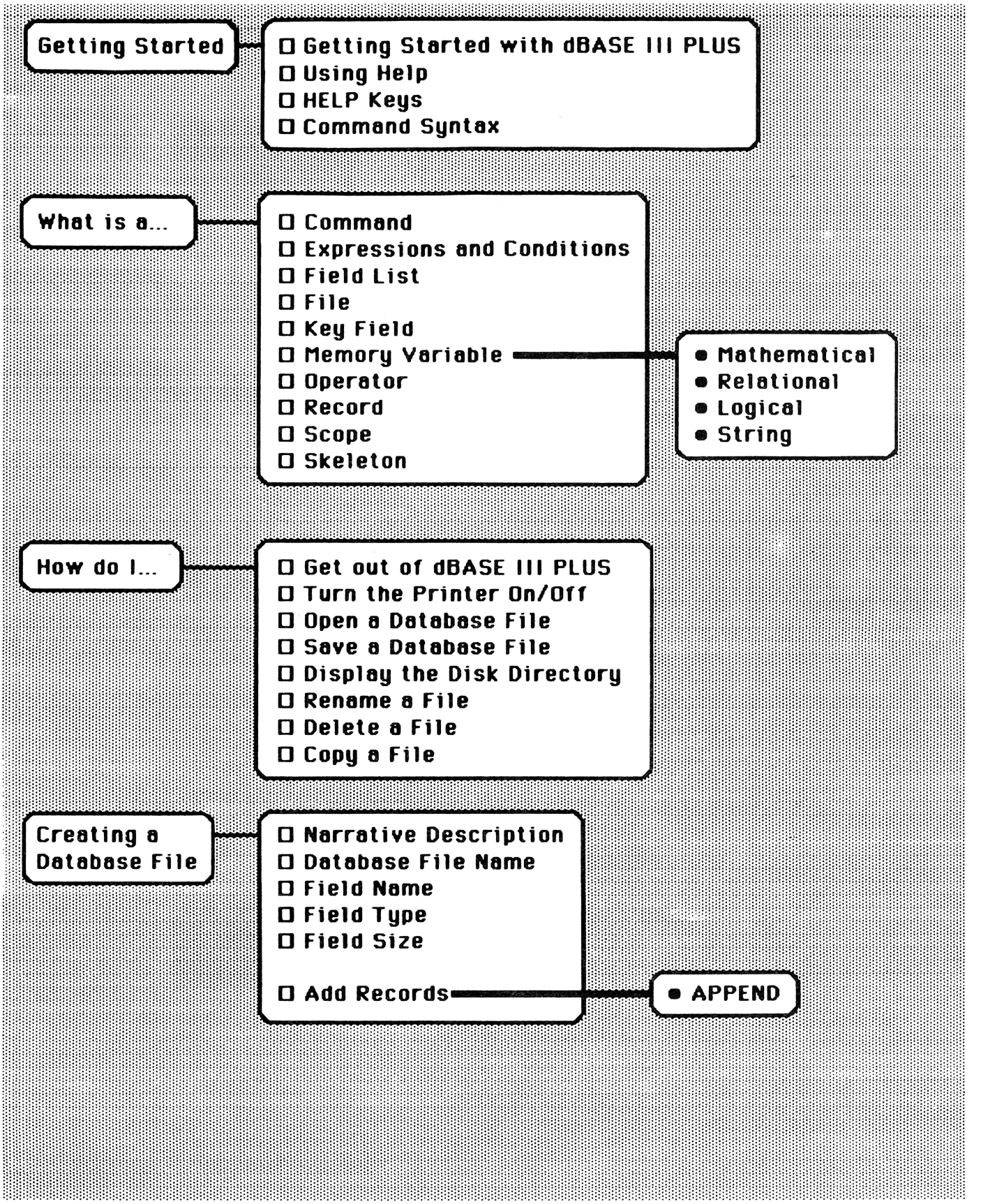
Using an existing database file describes the various commands that apply to manipulating database files in dBASE III PLUS. For example, there are five different ways to EDIT a record: BROWSE, CHANGE, EDIT, READ, and REPLACE. This section lists the different commands that can be used to accomplish a specific task with a full description and syntax of each command.

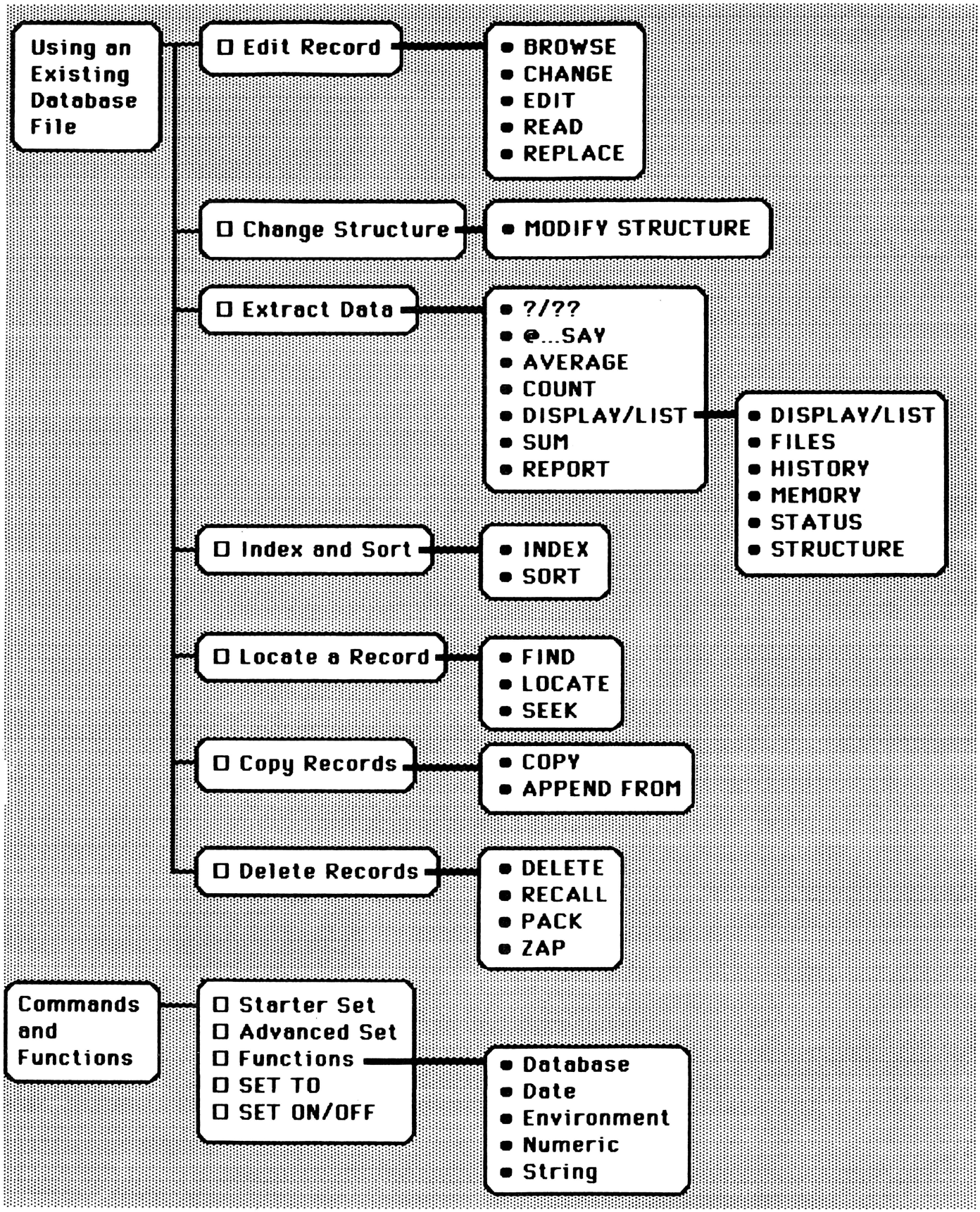
Commands and functions contains information on every available command and function in dBASE III PLUS. It provides a complete description of each command and its syntax.

The following is a map of HELP that shows its hierarchical structure and general layout. Use it as guide to understanding what HELP has to offer and as a reference.

Conclusion

As you can see from the HELP map, HELP is multi-leveled and quite complete. It is an excellent reference tool and an authoritative guide to the dBASE III PLUS command structures. If you've never thought of using HELP, try it. If you are supporting users in your office, teach them. It can save you both time and frustration.


Getting Started
Getting Started with dBASE III PLUS
Using Help
HELP Keys
Command Syntax
What is a...
Command
Expressions and Conditions
Field List
File
Key Field
Memory Variable
Operator
Record
Scope
Skeleton
Mathematical
Relational
Logical
String
How do I...
Get out of dBASE III PLUS
Turn the Printer On/Off
Open a Database File
Save a Database File
Display the Disk Directory
Rename a File
Delete a File
Copy a File
Creating a Database File
Narrative Description
Database File Name
Field Name
Field Type
Field Size
Add Records
APPEND


Using an Existing Database File
Edit Record
BROWSE
CHANGE
EDIT
READ
REPLACE
Change Structure
MODIFY STRUCTURE
Extract Data
?/??
@...SAY
AVERAGE
COUNT
DISPLAY/LIST
SUM
REPORT
DISPLAY/LIST
FILES
HISTORY
MEMORY
STATUS
STRUCTURE
Index and Sort
INDEX
SORT
Locate a Record
FIND
LOCATE
SEEK
Copy Records
COPY
APPEND FROM
Delete Records
DELETE
RECALL
PACK
ZAP
Commands and Functions
Starter Set
Advanced Set
Functions
SET TO
SET ON/OFF
Database
Date
Environment
Numeric
String

Protect Notes

by Richard Bunter and Karen Robinson

Protect is the stand-alone security utility that accompanies the dBASE Administrator. It allows you to define three types of security. First is login security, where you define a password for the user to enter before gaining access to the Administrator. Second is access control, where you assign to users file privileges limiting access to database files, fields, and applications. Third is data encryption, where you encrypt database files so only authorized users can access data.

For more information describing the theory and operation of Protect, refer to the "Network Administration" section in the *Programming with dBASE III PLUS* manual.

What follows is a series of usage tips applicable to using Protect.

Keep Track of Your Passwords

When setting up Protect, keep track of all passwords, group, and user names by printing a screen image of each user profile with **Shift-PrtSc**. There are several important reasons for doing this. First, Protect does not support listing user profile information. In order to display or edit a user profile, you must know the group name, user name, and user password. If this information is not available, then you will have to add the existing user as a new user.

A second important reason is the consequence of losing the Administrator password. Once entered, there is no mechanism to display it. If you lose it, you will not be able to enter Protect to add or edit profiles. If this happens, delete DBSYSTEM.DB and rebuild the Protect system using unencrypted versions of the database files. This may sound drastic, but it is important.

When you lose the Administrator password, you break the link between the encrypted database file and DBSYSTEM.DB. The Administrator password is embedded within each encrypted file linking access to it through the DBSYSTEM.DB file. You cannot regain access to encrypted database files by deleting the DBSYSTEM.DB and then re-encrypting them with a new Administrator password. Protect will not re-encrypt an encrypted database file. More importantly, if you erase the DBSYSTEM.DB file and set up a new one with a different password, you will not be able to USE the previously encrypted database files at all.

Remember, security implies responsibility. Keep track of your passwords.

Backup Encrypted Database Files Frequently

As a matter of course, make frequent backups of your unencrypted database files. One reason is described above. There is, in addition, another reason. A damaged, encrypted database file cannot be recovered. Often, however, an unencrypted database file can be recovered, in whole or in part. It is worth considering maintaining unencrypted backups of all encrypted database files as insurance against unlikely events.

Where DBSYSTEM.DB Should Be Located

In order for the Protect system to work, DBSYSTEM.DB must be in the default drive and directory when dBASE Administrator is invoked. If DBSYSTEM.DB is not present, the password login screen does not display and you are allowed access to the dBASE Administrator without password entry. You cannot, however, USE encrypted database files. Attempts to do so result in the error message "Database is encrypted."

If you are working in a network environment, copy DBSYSTEM.DB to the subdirectory that contains the Administrator files, and then set up a batch file so that every user accesses the dBASE Administrator from that subdirectory. This insures that each user goes through the password login screen.

.DBF Files in Several Subdirectories

Protect does not support subdirectories. If your database files are located in several different subdirectories, run Protect in each one copying, DBSYSTEM.DB, from one subdirectory to another. As you move between subdirectories, erase DBSYSTEM.DB from the last one. The following demonstrates exactly how to proceed.

1. Run Protect in the first subdirectory.
2. Copy the Protect file, DBSYSTEM.DB, to the next directory.
3. Delete DBSYSTEM.DB from the current directory.
4. Log the new directory.
5. Execute Protect again.
6. Repeat steps 2 - 4.

Single-User Security

If you need the security features Protect offers in a single-user environment, run the dBASE Administrator in place of dBASE III PLUS on your single-user machine.

Automatic Record Locking While EDITing

by Doug Haman

Locking a record while EDITing a database file in the dBASE Administrator requires that you press **Ctrl-O** to invoke the lock. The dBASE Administrator provides no automatic locking as it does with APPEND.

The following command file, Autolock.PRG, provides a mechanism for automatic record locking when EDITing records. When invoked, it places you at the beginning-of-file or the first record matching your search key, and then allows you to EDIT it. Once positioned in the database file, you can page between screen pages and records as you would normally in EDIT.

To set up, create Autolock.PRG and it's accompanying error-handling command file, Autoerr.PRG. Use the following syntax to execute Autolock.PRG.

```
DO Autolock WITH "<database file>", "<index file list>",;
                 "[<format file>]", "[<filter condition>]",;
                 "[<search key>]"
```

All PARAMETERS except for the database and index files are optional as indicated by the square brackets. If you omit a PARAMETER, send a null string in its place. For example, the following syntax is typical for a database and format file.

```
DO Autolock WITH "Data5000", "Data5000", "Data5000", "", ""
```

Additionally, Autolock.PRG assumes that index keys are character type. Pressing **Esc** or **Ctrl-End** exits you from Autolock, RETURNing control to the calling program or the dot prompt.

```
* Program ...: Autolock.PRG
* Author ....: Doug Haman
* Date ......: August 1, 1986
* Note(s) ...: Provides a facility for automatic record locking
*              while EDITing a database file. Assumes the database file
*              is INDEXed on a character key.
*
PARAMETERS dbfile, ndxfile, fmtfile, fltr, firstpos
* ---SET environment.
SET EXCLUSIVE OFF
SET TALK OFF
ON ERROR DO Autoerr
* ---Keys.
esc = CHR(12)
pgup = CHR(6)
uparrow = CHR(4)
ctrlend = CHR(14)
* ---Open files.
USE &dbfile. INDEX &ndxfle
SET FORMAT TO &fmtfile
* ---Define scopes.
SET FILTER TO &fltr
```

```
* ---Go to the first record.
IF "" = firstpos
   * ---Non-keyed.
   GO TOP
ELSE
* ---Keyed.
   SEEK firstpos
ENDIF
* ---Define more memvars.
mkey = 0
mlock = RLOCK()
* ---Edit'em cowboy.
DO WHILE .NOT. EOF()
   * ---Is it mine?
   IF mlock
      EDIT NEXT 1
      * ---Last key tells what to do next.
      DO CASE
      CASE CHR(MOD(READKEY(),256)) $ esc + ctrlend
         EXIT
      CASE CHR(MOD(READKEY(),256)) $ pgup + uparrow
         IF .NOT. BOF()
            SKIP -1
         ENDIF
      OTHERWISE
         SKIP
      ENDCASE
   ELSE
      SKIP
   ENDIF
   * ---Set and check the record lock.
   mlock = RLOCK()
   IF .NOT. mlock .AND. (.NOT. EOF())
      @ 23,00 CLEAR
      @ 23,00 SAY "Record is use by another"
      WAIT
   ENDIF
ENDDO
* ---Go home.
CLEAR ALL
ON ERROR
RETURN
* EOP Autolock.PRG
```

```
* Program ...: Autoerr.PRG
* Author ....: Doug Haman
* Date ......: August 1, 1986
* Note(s) ...: Error routine for Autolock.PRG.
*
locked = 109      && SKIPping to a locked record.
eof    = 4        && End-of-file encountered.
* ---Interesting trick.
cmd = IIF(ERROR() = locked, "mlock = .F.",;
          IIF(ERROR() = eof, "GO TOP",;
          "? ERROR(),MESSAGE()"))
&cmd
RETURN
* EOP Autoerr.PRG
```

Multiple Across Reporting Using LABEL FORM

by Karen Robinson

LABEL FORM in combination with a series of @...SAY statements can be used to generate paginated column reports with headers and footers. You use LABEL FORM to print records in the form of labels and @...SAY to print the headers and footers on each page. This type of report is often used for client or membership lists, although there may be other applications for it. The report appearance is professional and requires very little programming. The following is an example of what a typical report of this type looks like.

```
Page 1
08/01/86

                    Membership List

Ashton-Tate           John Doe            Jane Smith
20101 Hamilton Ave.   123 State St.       456 Beach Blvd.
Torrance, CA 90250    Lawndale, CA 90233  Suite 204
                                          Los Angeles, CA 90066
                              .
                              .
                              .
                      -- The List Group --
```

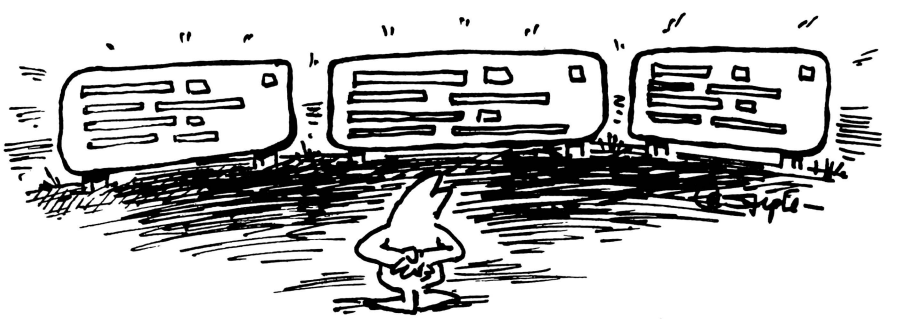
The following program, Multirpt.PRG, prints such a report. To set up, first CREATE a LABEL FORM with the appropriate formatting. Be sure to define the number of labels across. Second, create the command file, Multirpt.PRG replacing the filenames with your own, repositioning the headers and footers as necessary, and changing the numeric argument of LABEL FORM NEXT <number per page> to the appropriate value for the number of records to be printed per page.

The number of labels per page depends on the number of columns you define and the number of lines available after the headers and footers print. The following formula shows how to calculate it.

```
((<lines per page> - <top margin>;
+ <bottom margin>) / <lines per label>) * <labels across>
```

The following pseudo-code describes the mechanics of Multirpt.PRG.

1. Set up the working environment.
   - Define environmental SETtings.
   - Open your database file.
   - Initialize page counter variable to zero.
   - Direct output to the printer.
2. Print the report for all records in your database file.
   - Print header(s) on a new page.
   - Print one page of data using LABEL FORM/NEXT.
   - Print footer(s) at bottom of page.
   - Increment page counter.
3. Restore the environment.
   - Advance the printer to a new page.
   - Direct output to the screen.
   - Close the database.
   - RETURN to calling program or the dot prompt.



```
* Program..: Multirpt.PRG
* Author...: Karen Robinson
* Date.....: August 1, 1986
* Notes....: This program prints multiple records in columns
*            across the page. Anything delimited with angle
*            brackets needs to be replaced with an appropriate
*            filename or value.
*
SET TALK OFF
USE <database file>
page = 1
SET DEVICE TO PRINT
DO WHILE .NOT. EOF()
   * ---Print header at top of the page.
   @ 2, 1 SAY "Page"
   @ 2, 6 SAY STR(page,2)
   @ 3, 1 SAY DATE()
```

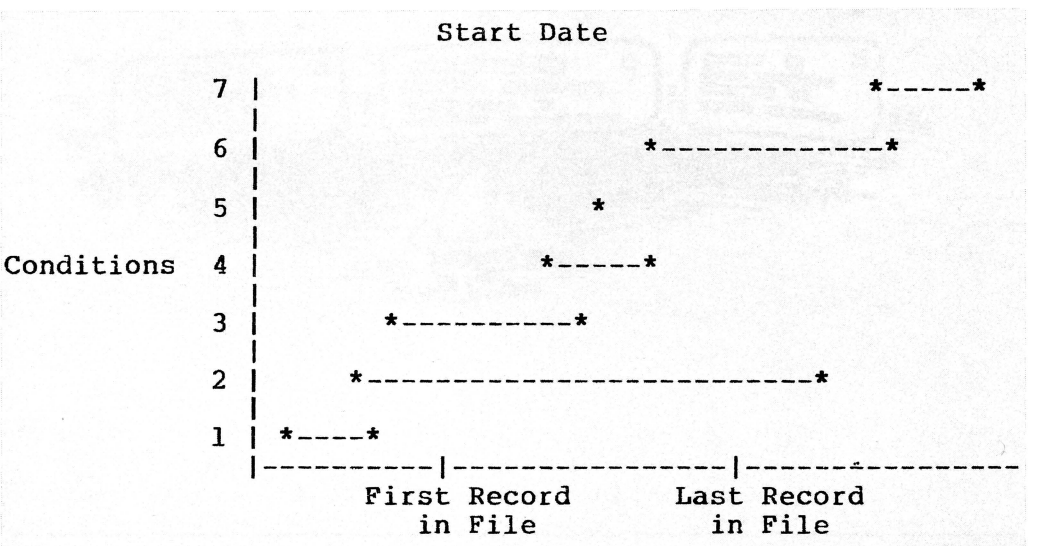
```
   @ 5,10 SAY "This is the header"
   @ 7,0
   * ---Print one page of data.
   LABEL FORM <label file> NEXT <number per page> TO PRINT
   * ---Print Footer.
   @ 60, 2 SAY "This is the footer"
   * ---Increment page counter.
   page = page + 1
ENDDO
* ---Restore environment.
EJECT
SET DEVICE TO SCREEN
CLOSE DATABASES
RETURN
* EOP Multirpt.PRG
```

Searching for the First Date Within a Range

by Richard Goepel and Karen Robinson

Many database applications require date fields. Dates of birth, hire, transactions, and appointments are just a few examples. Since chronological order is an important method of accessing collections of data, applications typically generate reports in date order.

A single date search requires only a test for the existence of the specified date in the database file. dBASE III PLUS provides many suitable searching techniques for a single date, though FIND and SEEK are the quickest. Often, however, you want to report on a range of dates but do not know the exact starting date. Furthermore, the date range you specify may or may not exist in the database file. The starting date can be in one of seven places in the database file. The following graph illustrates these conditions:



These seven conditions divide into four main categories.

1. The starting and ending dates are the same (condition 5).
2. The starting and ending dates are outside the date ranges of the database file (conditions 1 and 7).
3. The starting date is less than the date stored in the first record of the database file (conditions 2 and 3). In either of these situations, the first record of the database file is the starting date.
4. The starting date is embedded within the range of values stored in the first and last records of the database file (conditions 4 and 6).

For small to medium database files, these evaluations can be handled by any of the output commands that use the FOR clause, such as REPORT FORM FOR <condition>, DISPLAY FOR <condition>, and LIST FOR <condition>. However, for large database files, using the FOR clause can be slow.

The following program, Datefind.PRG, is a search routine that moves the record pointer to the first date in a specified range, if the date exists. The first part of the subroutine determines if the date range specified is within the boundaries of the dates stored in the database file (conditions 1 and 7). If the date range is not out of bounds, then conditions 2 and 3 are tested for. In this case, the starting date is the first record of the database file and the program stops. Otherwise, the starting and ending dates are tested to see if they are equal. If they are, a SEEK is done for a single date. If they are not, conditions 4 and 6 are tested for. This is done by searching for the year first, then the month within that year, then the day within that month and year.

This routine can be used as a procedure within a custom program or run just prior to a command that uses the WHILE clause to process a group of records. An example is:

```
REPORT FORM Yourfile WHILE Date_field >= start_date;
                     .AND. Date_field <= end_date
```

To use the Datefind.PRG, INDEX your database file ON the character conversion of the date field in the format "YYYYMMDD." The following command syntax demonstrates how:

```
INDEX ON STR(YEAR(<Date_field>),4) + ;
         SUBSTR(DTOC(<Date_field>),1,2) + ;
         SUBSTR(DTOC(<Date_field>),4,2) TO <your index>
```

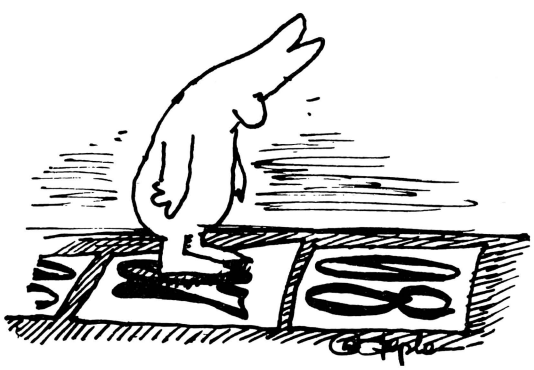
One logical and two date variables are passed as parameters to the date search subroutine: "nofind," "start_date," and "end_date." Once it is determined that the starting date variable is embedded within the database file (conditions 4, 5, or 6), the date variables are converted into the same format as the index key ("YYYYMMDD"). The logical variable, "nofind", keeps track of whether or not a date was found. Datefind.PRG returns a true value (.T.), if a date was not found.

Main.PRG demonstrates how to set up a program to accept and test the validity of the two date variables (start_date and end_date) before executing the date search subroutine. In this sample program, if either of the variables are left blank, the program returns to the dot prompt.

As a side note, once a value is entered into a date field or variable, the only way to blank it out is to use **Ctrl-Y** at the first position.

```
* Program ... : Main.PRG
* Author .... : Richard Goepel
* Date ...... : August 1, 1986
* Note(s) ... : This program accepts the starting and ending
*               dates and then calls Datefind.PRG
*
*               Yourfile   ::= Database file that contains the
*                              dates to be searched.
*               Yourindex  ::= Database file indexed on the
*                              character conversion of date.
*               Date_field ::= Date field to be searched.
*
USE Yourfile INDEX Yourindx
DO WHILE .T.
   STORE CTOD(" / / ") TO start_date, end_date
```

```
   @ 10,10 SAY "Enter start date" GET start_date
   @ 11,10 SAY "Enter end date " GET end_date
   READ
   IF DTOC(start_date) = " / / " .OR.;
      DTOC(end_date) = " / / "
      EXIT
   ENDIF
   * ---Test for a valid date range.
   IF start_date > end_date
      @ 20,19 SAY "Start date cannot be greater than " + ;
                  "end date."
      LOOP
   ENDIF
   * ---If nofind comes back true, a date was not found.
   nofind = .F.
   mend_date = end_date
   * ---Date search subroutine called from here.
   DO Datefind WITH start_date, end_date, nofind
   IF nofind
      @ 20,22 SAY "Date range specified cannot be found"
   ELSE
      * ---Convert start_date and end_date back to a date.
      start_date = Date_field
      end_date = mend_date
      EXIT
   ENDIF
ENDDO
* EOP Main.PRG
```



```
* Program ... : Datefind.PRG
* Author .... : Richard Goepel
* Date ...... : August 1, 1986
* Note(s) ... : This routine moves the record pointer to the
*               first date within a specified date range.
*
PARAMETERS start_date, end_date, nofind
* ---Test to see if the starting and ending dates fall within
* ---the date range of the database file.
GO TOP
IF start_date < Date_field
   IF end_date < Date_field
      nofind = .T.
```

```
   ENDIF
   RETURN
ENDIF
GO BOTTOM
IF start_date > Date_field
   nofind = .T.
   RETURN
ENDIF
* ---Convert the date variables to the format of the key field,
* ---"YYYYMMDD".
start_date = STR(YEAR(start_date),4) +;
             SUBSTR(DTOC(start_date),1,2) +;
             SUBSTR(DTOC(start_date),4,2)
end_date = STR(YEAR(end_date),4) +;
             SUBSTR(DTOC(end_date),1,2) +;
             SUBSTR(DTOC(end_date),4,2)
*
* ---Search for a single date.
IF start_date = end_date
   SEEK start_date
   IF .NOT. FOUND()
      nofind = .T.
   ENDIF
   RETURN
ENDIF
* ---Search for a matching year.
SEEK SUBSTR(start_date,1,4)
IF .NOT. FOUND()
   nofind = .T.
   * ---Repeat search for years less or equal to the year of
   * ---end_date.
   DO WHILE SUBSTR(start_date,1,4) <= SUBSTR(end_date,1,4)
      * ---Increment the year by 1 and search again.
      start_date = STR(VAL(SUBSTR(start_date,1,4)) + 1,4);
                   + SUBSTR(start_date,5,4)
      SEEK SUBSTR(start_date,1,4)
      IF FOUND()
         nofind = .F.
         * ---If the year is found, start_date becomes January 1
         * ---of that year.
         start_date = SUBSTR(start_date,1,4) + "0101"
         EXIT
      ENDIF
   ENDDO
ENDIF
* ---If year is not found, end the routine.
IF nofind
   RETURN
ENDIF
* ---Search for the year and month of start_date.
SEEK SUBSTR(start_date,1,6)
IF .NOT. FOUND()
   nofind = .T.
   * ---Repeat search for month less than 13.
   DO WHILE VAL(SUBSTR(start_date,5,2)) < 13
      * ---Increment month by 1 and search again.
      start_date = STR(VAL(SUBSTR(start_date,1,6)) + 1,6) +;
                   SUBSTR(start_date,7,2 )
      SEEK SUBSTR(start_date,1,6)
      IF FOUND()
         nofind = .F.
         * ---Replace date string with 1st day of month.
         start_date = SUBSTR(start_date,1,6) + "01"
```

```
         EXIT
      ENDIF
   ENDDO
ENDIF
* ---If month is not found, end the routine.
IF nofind
   RETURN
ENDIF
* ---Year and month are found. Now, search for day.
SEEK start_date
IF .NOT. FOUND()
   nofind = .T.
   DO WHILE VAL(SUBSTR(start_date,7,2)) < 32
      * ---Increment day by 1 and search again.
      start_date = STR(VAL(start_date) + 1,8)
      SEEK start_date
      IF FOUND()
         nofind = .F.
         EXIT
      ENDIF
   ENDDO
ENDIF
RETURN
* EOP Datefind.PRG
```

Conclusion

Datefind.PRG is most effective when used on large database files where you do not know the exact starting date. This same kind of routine can be used to search character or numeric data types as well.

Importing Delimited Files with No Delimiters in dBASE III

by John Thomas Regen

In dBASE II, you can import an ASCII text file separated by commas and no delimiters bounding character strings, using APPEND FROM DELIMITED. This is not supported in versions 1.0 and 1.1 of dBASE III, however. If you do not have the Developer's Release or dBASE III PLUS, you can use the following command file, Commas.PRG, to import a text file with no delimiters into a dBASE III database file.

Commas.PRG, takes an ASCII text file separated with commas or any specified character, COPYs it into a temporary dBASE III file, and then using sub-string searches, builds target dBASE III database file records. There are several significant limitations to this implementation. First, you can only import records that have a record length of 254 characters or less. Second, the target database file can only have character fields. If you are importing data that is date, numeric, or logical type, convert the target fields to the appropriate data type with MODIFY STRUCTURE after you have imported the data.

Given these limitations, Commas.PRG, gives dBASE III, version 1.0 and 1.1 users the ability to import small data files typical of mail merge applications, into a dBASE III database file.

To set up the importation process, follow these steps.

1. Create a temporary database file called Temp.DBF with the following structure.

Field	Field Name	Type	Width	Dec
1	Onlyfield	Character	254	

2. Create the target database file, Newfile.DBF, with the number and length of character fields (named Field1, Field2, etc.) as follows.

Field	Field Name	Type	Width	Dec
1	Field1	Character	10	
2	Field2	Character	25	
	.			
	.			
N	FieldN	Character	12	

   Make sure that each field in Newfile.DBF corresponds to the source field length. Don't forget to account for the decimal place and sign if the field is numeric. For date fields, account for the embedded slashes if the source field has them.

3. Rename the delimited text file to import to Test.TXT.

4. Create Commas.PRG using as many variables as you have fields.

```
* Program ...: Commas.PRG
* Author ....: John Thomas Regen
* Date ......: August 1, 1986
* Note(s) ...: Imports an undelimited, comma separated text file
*              into a dBASE III version 1.0 or 1.1 database file.
*              Note that the maximum import record size is 254
*              characters.
*
SET SAFETY OFF
SET TALK OFF
* ---Open files.
SELECT 1
USE Newfile
ZAP
SELECT 2
USE Temp
ZAP
APPEND FROM Test.TXT SDF
* ---Process all records in the source database file.
DO WHILE .NOT. EOF()
   * ---For each field in your database file define a variable.
   * ---If you have 30 fields you should have variables up to
   * ---var30.
   STORE " " TO var1, var2, var3, var4, var5, var6
   STORE TRIM(onlyfield) TO trim_field
   sub_ptr = 1
   DO WHILE LEN(trim_field) > 0
      * ---Convert sub_ptr to sub trimming leading blanks.
      sub     = STR(sub_ptr,10-AT("-",STR(-sub_ptr)))
      var&sub = SUBSTR(trim_field,1,AT(",",trim_field)-1)
      SELECT Newfile
      * ---If this is the first target field,
      * ---create a new target record.
```

```
      IF sub_ptr = 1
         APPEND BLANK
      ENDIF
      * ---Write to the current target field.
      REPLACE field&sub WITH var&sub
      SELECT Temp
      * ---Get another block of source.
      var_len = LEN(var&sub)
      IF var_len + 2 > LEN(trim_field)
         trim_field = ""
      ELSE
         trim_field = SUBSTR(trim_field,var_len + 2)
         sub_ptr = sub_ptr + 1
      ENDIF
   ENDDO
   * ---Get next source record.
   SKIP
ENDDO
RETURN
* EOP Commas.PRG
```

Advanced Sorting Primitives for dBASE TOOLS FOR C

by Robert Boies

This article continues the series on sorting and searching. The set of primitives presented here form the basic set of commonly used searching and sorting algorithms. This article focuses on the use of the dBASE TOOLS FOR C package as the array handling mechanism. In the first article, I defined a primitive as a data independent procedure that performs a single task and returns one or two values. The primitive sits at the lowest level of functional hierarchy in the system. To this level is added the metaprimitive. The metaprimitive is defined as a procedure that performs a simple task composed of basic primitives. It differs from higher level functions by acting only as a method for grouping lower level primitives together.

The literature is consistent in its definitions of the low-level primitives. It is not consistent when defining which algorithms make up the metaprimitives or which comprise the high-level sorts. There are three basic sorting tools that the literature seems to agree on as metaprimitives. These are: selecting, inserting, and partitioning.

To facilitate your understanding of these new metaprimitives, a general review of the low-level primitives is presented first. Before going further in the discussion, it is suggested that you build a procedure file with the procedures listed below.

The New Primitives

Discussion of Improvements

The primitives in this article have been presented in the last article in this series. At the time, there were intentional sacrifices in performance of the algorithms to provide clarity in presentation. The code presented in this article was rewritten to be more efficient. If you find difficulty with the approach of the new primitives, I suggest you reference the previous article for a better understanding. Keep in mind, the metaprimitives are based on the low-level primitives that follow.

There are some general improvements to note which are common to all of the algorithms:

1. Array names are now passed in macro variables.
2. Whenever possible, the new dBASE III PLUS command syntax has been used to reduce the overall parsing time for the primitive operation.

The actual code follows a listing of program headers which define the parameters passed and the local variable set. These headers are normally presented at the top of each program, but have been grouped together here to provide clarity for the overall use of the primitives. I suggest you use this as a "quick reference" and study the list of functions.

Program Headers

```
PROCEDURE C_sequen
PARAMETERS array_name, array_type, begin, end, comp_op, key,;
           elem_ptr, rtrn_code
*********************************************************************
* Performs a sequential search on the contents of an array and      *
* returns a value satisfying the "comp_op" condition.               *
*********************************************************************
array_name  := Name of the array to be searched
array_type  := Type of array, := = "c"|"d"|"n"
start       := Starting point of the search
stop        := Ending point of the search
comp_op     := Comparison operator, := =
               "<"|">"|"="|"<="|">="|"<>"
key         := Key value being searched for
elem_ptr    := Element pointed to after completion of the search
rtrn_code   := Return variable containing completion code
```

```
PROCEDURE C_binary
PARAMETERS array_name, array_type, low, high, elem_ptr, key,;
           comp_op, rtrn_code
*********************************************************************
* Performs a binary search on the contents of an array and          *
* returns a value satisfying the "comp_op" condition.               *
*********************************************************************
array_name  := Name of the array to be searched
array_type  := Type of array, := = "c"|"d"|"n"
low         := Starting element to use in the search
high        := Ending element to use in the search
key         := Key value being searched for
elem_ptr    := Element pointed to after completion of the search
comp_op     := Comparison operator, := =
               "<"|">"|"="|"<="|">="|"<>"
rtrn_code   := Return variable containing completion code
```

```
PROCEDURE C_exchan
PARAMETERS src_array, tgt_array, array_type, src_ptr, tgt_ptr,;
           rtrn_code
*********************************************************************
* Exchanges two elements of an array or two elements between        *
* arrays.                                                           *
*********************************************************************
src_array   := Name of the source array
tgt_array   := Name of the target array
array_type  := Type of array, := = "c"|"d"|"n"
src_ptr     := Pointer to the source array
tgt_ptr     := Pointer to the target array
rtrn_code   := Return variable containing completion code
```

```
PROCEDURE C_select
PARAMETERS src_array, tgt_array, array_type, start, stop,;
           tgt_ptr, nxt_elem, elem_ptr, comp_op, rtrn_code
*********************************************************************
* Selects the next logical item in a series from "src_array"        *
* based on the value of an item pointed to in "tgt_array" by        *
* "tgt_ptr" .AND. the "comp_op" condition, returns the next         *
* element, "nxt_elem" of the series and location "elem_ptr"         *
* the source array.                                                 *
*********************************************************************
src_array    := Name of the source array
tgt_array    := Name of the target array
array_type   := Type of array, := = "c"|"d"|"n"
start        := Starting element to use in the search
stop         := Ending element to use in the search
tgt_ptr      := Element of series pointed to in target array
nxt_elem     := Next item in the series selected from "src_array"
elem_ptr     := Pointer to next element of series in "src_array"
comp_op      := Comparison operator, := =
                "<"|">"|"="|"<="|">="|"<>"
rtrn_code    := Return variable containing completion code
```

```
PROCEDURE C_convrt
PARAMETERS key_in, array_type, key_str, rtrn_code
*********************************************************************
* Performs type conversion of a variable passed in "key_in" to      *
* character.                                                        *
*********************************************************************
key_in       := Input data value
array_type   := Type of array, := = "c"|"d"|"n"
key_str      := String representation of the data
rtrn_code    := Return code containing completion code
GLOBALS      :
g_precsn     := Precision for numerics
g_scale      := Scale for numerics
```

```
PROCEDURE C_compar
PARAMETERS array1, array2, elem_ptr1, elem_ptr2, array_type,;
           comp_op, rtrn_code
*********************************************************************
* Compares two array elements according to the comparison specs     *
* stored in "comp_op" and sets "rtrn_code" to 1 if true or -1 if    *
* false.                                                            *
*********************************************************************
array1       := Name of the first array
array2       := Name of the second array
elem_ptr1    := Pointer first element
elem_ptr2    := Pointer second element
array_type   := Type of array, := = "c"|"d"|"n"
comp_op      := Comparison operator, := =
                "<"|">"|"="|"<="|">="|"<>"
rtrn_code    := Return variable containing completion code
```

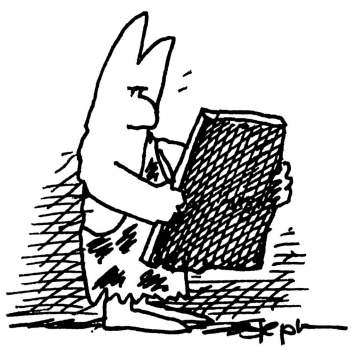
```
PROCEDURE C_insert
PARAMETERS src_array, tgt_array, start, stop, array_type,;
           src_ptr, max_elem, comp_op, rtrn_code
*********************************************************************
* Inserts the next logical element in a series into the             *
* "tgt_array" picked from a group of elements in "src_array".       *
*********************************************************************
src_array    := Name of the source array
tgt_array    := Name of the target array
start        := Starting point of the search
stop         := Ending point of the search
array_type   := Type of array, := = "c"|"d"|"n"
src_ptr      := Pointer to the source array
max_elem     := Last element in the array
comp_op      := Comparison operator, := =
                "<"|">"|"="|"<="|">="|"<>"
rtrn_code    := Return variable containing completion code
```

```
PROCEDURE C_partit
PARAMETERS array_nme, max_elem, start, stop, lft_lim_lft_lst,;
           rgt_lim_lft_lst, lft_lim_rgt_lst,;
           rgt_lim_rgt_lst, rtrn_code
*********************************************************************
* Partitions an array into two integral pieces based on the         *
* value stored in a given element (start).                          *
*********************************************************************
array_nme          := Name of the array
max_elem           := Number of elements in the array
start              := Pointer to element to start with
stop               := Pointer to element to stop with
lft_lim_lft_lst    := Lefthand limit of LEFT list
rgt_lim_lft_lst    := Righthand limit of LEFT list
lft_lim_rgt_lst    := Lefthand limit of RIGHT list
rgt_lim_rgt_lst    := Righthand limit of RIGHT list
rtrn_code          := Return variable containing completion code
```

Primitive Procedures

```
PROCEDURE C_sequen
PARAMETERS array_name, array_type, start, stop, comp_op, key,;
           elem_ptr, rtrn_code
*
elem_ptr = start + IIF(stop > start, - 1, 1)
DO WHILE (elem_ptr <> stop) .AND.;
          (key &comp_op. c_&array_type._result)
   elem_ptr = elem_ptr + 1
   CALL cfunc WITH "GETARRAY &array_name.," + STR(elem_ptr)
ENDDO
rtrn_code = c_status
RETURN
* EOP C_sequen
```

```
PROCEDURE C_binary
PARAMETERS array_name, array_type, low, high, elem_ptr, key,;
           comp_op, rtrn_code
*
found = .F.
DO WHILE ((.NOT. found) .AND. (low < high))
   CALL cfunc WITH "GETARRAY &array_name.," +;
                   STR(INT((low + high) / 2))
   IF key = c_&array_type._result
      found = .T.
      LOOP
   ELSE
      new_val = IIF(key &comp_op. c_&array_type._result,;
                    "high", "low")
      &new_val = INT((low + high) / 2) + 1
   ENDIF
ENDDO
IF .NOT. found
   rtrn_code = -1
ENDIF
elem_ptr = IIF(rtrn_code = -1, -1, INT((low + high) / 2))
RETURN
* EOP C_binary
```



```
PROCEDURE C_compar
PARAMETERS array1, array2, elem_ptr1, elem_ptr2, array_type,;
           comp_op, rtrn_code
*
CALL cfunc WITH "GETARRAY &array1., " + STR(elem_ptr1)
temp = c_&array_type._result
CALL cfunc WITH "GETARRAY &array2., " + STR(elem_ptr2)
rtrn_code = IIF(temp &comp_op. c_&array_type._result, 1, -1)
RETURN
* EOP C_compar
```

```
PROCEDURE C_exchan
PARAMETERS src_array, tgt_array, array_type, src_ptr,;
           tgt_ptr, rtrn_code
CALL cfunc WITH "GETARRAY &src_array., " + STR(src_ptr)
temp = c_&array_type._result
CALL cfunc WITH "GETARRAY &tgt_array., " + STR(tgt_ptr)
CALL cfunc WITH "PUTARRAY &tgt_array., " + STR(tgt_ptr) +;
                ", " + temp
CALL cfunc WITH "PUTARRAY &src_array., " + STR(src_ptr) +;
                ", " + c_&array_type._result
rtrn_code = c_status
RETURN
* EOP C_exchan
```

```
PROCEDURE C_convrt
PARAMETERS key_in, array_type, c_precsn, c_scale, key_str, rtrn_code
key_str = IIF(array_type = "N",;
              STR(c_n_result, c_precsn, c_scale),;
              IIF(array_type="D",DTOC(c_d_result),;
                  c_c_result))
IF LEN(key_str) = 0
   rtrn_code = -1
ENDIF
RETURN
* EOP C_convrt
```

The Metaprimitives

Selection Sorts

Selection sorting is the process of selecting the next higher or lower value in a series. When creating a sort model using selection, it is possible to pass a pointer to an array element, or the value of the element, back to the program that calls the selecting procedure. I have chosen to focus on passing pointers rather than values. It is more efficient to put data into an array and leave it alone, than to move the data within the array. This approach minimizes internal data movement and does not have to include any code to move data. When building a sorted order, an array of pointers is constructed to point to the locations of the data in the data array.

```
PROCEDURE C_select
PARAMETERS src_array, tgt_array, array_type, start, stop,;
           tgt_ptr, nxt_elem, elem_ptr, rtrn_code
*
CALL cfunc WITH "GETARRAY &tgt_array." + ", " + STR(tgt_ptr)
elem_ptr = start
DO C_sequen WITH src_array, array_type, start, stop, comp_op,;
                 elem_ptr, c_&array_type._result, rtrn_code
CALL cfunc WITH "GETARRAY &src_array." + ", " + STR(elem_ptr)
nxt_elem = c_&array_type._result
RETURN
* EOP C_select
```

Insertion Sorts

The insertion sort is the inverse of the selection sort. As with the selection sort, the code is written using the mechanism of pointer manipulation. The reasons for using pointers become clearer as we go along. Since there is dislocation in the array when an insertion is performed, it makes more sense to move only pointers instead of actual data.

The essential difference between the select and insert primitives is that selection always returns the object of insertion, where insertion always returns the position of selection. Consider another way of looking at it. Insert returns the position of insertion, the location to create an opening in the target array. Select, on the other hand, returns the key or pointer to be inserted; in other words, what to place in the new opening.

A number of the primitives can be combined with the insertion technique to build new high level sorts. For instance, examine the action of combining the insertion mechanism with the sequential search and use the exchange primitive to swap the current element pointer in the bucket with a pointer array element. The element swapped out can then be used as the lookup into the data array on the next pass, or placed in an unused position in the array. This is actually the design of one of the accepted high-level sort mechanisms, linear insertion with exchange.

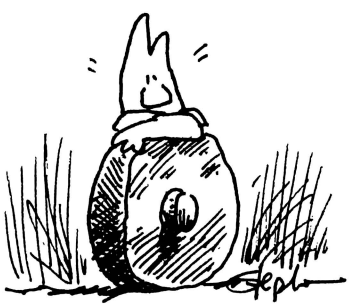
```
PROCEDURE C_insert
PARAMETERS src_array, tgt_array, start, stop, array_type,;
           src_ptr, max_elem, comp_op, rtrn_code
*
CALL cfunc WITH "GETARRAY &src_array., " + STR(src_ptr)
key = c_&array_type._result
DO C_sequen WITH tgt_array, array_type, start, stop, comp_op,;
                 key, elem_ptr, rtrn_code
n = max_elem
DO WHILE (n > elem_ptr)
   DO C_exchan WITH tgt_array, tgt_array, n, n - 1, array_type,;
                    rtrn_code
   n = n - 1
ENDDO
key_str = space(254)
DO C_convrt WITH key, array_type, key_str, rtrn_code
key_str = trim(key_str)
CALL cfunc WITH "PUTARRAY &tgt_array., " + STR(elem_ptr + 1) +;
                ", " + key_str
```

```
CALL cfunc WITH "PUTARRAY &src_array., " + STR(src_ptr) +;
                ", " + SPACE (0)
RETURN
* EOP C_insert
```

Partitioning

Partitioning a set of data is a concept you use all the time with a database file. When you have a database file that contains a set of records and wish to view these records in groups, you build an index file and allow dBASE III PLUS to group them for you. In the arrays there is no index mechanism available, so you have to construct one. It is more efficient and simple to first partition the larger set into separate smaller subsets. Other sort and search facilities, like the primitives, will operate better on these small sets than on the entire array.

The partitioning mechanism is started by selecting a seed. This seed is usually the first element of the unsorted array. The key values stored in higher numbered array elements of the data array are compared to the value of this seed according to the comparison operator passed to it in "comp_op." As an example, if the object of the sort procedure is ascending order, the keys are compared to the seed until a key of less than or equal value is found. The first partition of the array is now defined as starting at the first key and ending at the current element minus one. The process is repeated until the end of the array is reached.



This operation partitions the array into segments. Each segment can be treated separately and sorted internally. They can be searched according to the original order, and the partition pointer array can be sorted. Thus, partitioning makes the array easier to manage.

If the partition pointer array is sorted first, the total number of key lookups is substantially reduced. Accordingly, you need only compare a lookup key to the first and last element in the actual data array. If the key is found to fall between these values, this is the partition where the key will be found if it exists. If it does not, the search proceeds to the next partition pointed to by the partition pointer array. This continues repeating until the total number of partitions have been exhaustively searched, or the key is found.

If you notice a similarity between the partitions and pages of an index file, go to the head of the class! This concept is similar to the one used when dBASE builds the index pages for the index file. The major difference is that the partitions are all fixed length in dBASE indexes, whereas the partitions in the array may vary in size.

```
PROCEDURE C_partit
PARAMETERS Array_nme, max_elem, start, stop, lft_lim_lft_lst,;
           rgt_lim_lft_lst, lft_lim_rgt_lst,;
           rgt_lim_rgt_lst, rtrn_code
*
* ---Start the process by getting the value of the starting
* ---element to be used as the pivot.
CALL cfunc WITH "GETARRAY &array_nme.," + STR(start)
pivot = c_&array_type._result
lft = start
rgt = stop + 1
* ---Scan from both ends, one element at a time, as long as the
* ---inner endpoints of the scanned section are distinct from
* ---each other.
DO WHILE (lft < rgt)
   rgt = rgt - 1
   CALL cfunc WITH "GETARRAY &array_nme.," + STR(rgt)
   con_curr_elem = c_&array_type._result
   *
   * ---Scan to the right.
   DO WHILE (con_curr_elem > pivot)
      CALL cfunc WITH "GETARRAY &array_nme.," + STR(rgt)
      con_curr_elem = c_&array_type._result
      rgt = rgt - 1
   ENDDO
   lft = lft + 1
   CALL cfunc WITH "GETARRAY " + Array_nme + STR(lft)
   con_curr_elem = c_&array_type._result
   *
   * ---Scan to the left.
   DO WHILE ((con_curr_elem < pivot) .AND. (lft < rgt))
      CALL cfunc WITH "GETARRAY &array_nme.," + STR(lft)
      con_curr_elem = c_&array_type._result
      lft = lft + 1
   ENDDO
ENDDO
* ---Set up the partition definition pointers
* ---(limits of the lists) and return to the calling program.
ptr = right
lft_lim_lft_lst = start
rgt_lim_lft_lst = ptr - 1
lft_lim_rgt_lst = ptr + 1
rgt_lim_rgt_lst = stop
RETURN
* EOP C_partit
```

Conclusion

In this article, the concepts and mechanisms of the metaprimitives of selecting, inserting, and partitioning an array were introduced. A discussion of some of the theory of operation of the array mechanisms of dBASE TOOLS FOR C was covered relating to creating sorted datasets. Next month, I will conclude the discussion of sorting and searching with examples of high level sorts.

USAGE TIPS

dBASE III PLUS

by Christopher White

SET RELATION TO

Throughout the documentation for dBASE III PLUS, there is some ambiguity about the various ways you can link database files with SET RELATION TO.

SET RELATION TO relates two database files in three different ways.

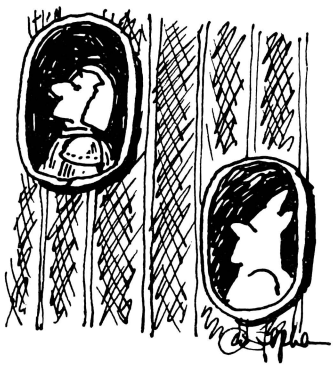
1. You can relate two database files with a key expression allowing you to create connections between database files based on value. It is important to stress that you are not confined to a single common field. The linking expression can be any legitimate dBASE III PLUS expression. To work, the key expression must return a value from the source work area that matches a value in the master index file of the target work area. Because the relation is created by an expression, you can take two dissimilar database file structures that contain common values and create a relation between them. In exactly the same way, you can create a relation between database files that have common fields. In both cases, the mechanism is the same: the relation is created by an expression.

2. Another way to relate database files is by record number. This type of relation allows you to link two database files having no common field values but have a one-to-one relation based on physical position. Record one in the source corresponds to record one in the target file and so on. Generally, you will use this formulation to create a virtual database file where you need a number of fields greater than the dBASE III PLUS limit of 128 per database file.

   When the record pointer moves in the source database file, the record pointer in the target moves to the same record number as the source record. For example,

```
* —-Setup and syntax with a physical relation.
SELECT B
USE File2
SELECT A
USE File1 INDEX File1
SET RELATION TO RECNO( ) INTO File2
* —-List records from File1 and File2.
DO WHILE .NOT. EOF( )
        ? Field1, Field2, File2->Field1, File2->Field2
        * —-Advance File1 and File2 pointer.
        SKIP
ENDDO
*
* —-Setup and syntax with no relation.
SELECT B
USE File2
SELECT A
USE File1 INDEX File1
* —-List records from File1 and File2.
DO WHILE .NOT. EOF( )
```

```
            ? Field1, Field2, File2->Field1, File2->Field2
            * —-Advance the File1 pointer.
            SKIP
            * —-Advance the File2 pointer.
            recpos = RECNO( )
            SELECT B
            GO recpos
            SELECT A
ENDDO
```



In order to build various index sort orders, put all of your key fields in the master database file as the diagram below indicates. This is important since a relation based on RECNO( ) precludes the use of an index file in the target work area.

```
Master ——-> Sub1 ———> Sub2
(Keys)      (Non-keys)   (Non-keys)
```

3. The last type of relation is a physical link between database files based on a numeric expression. If the linking expression evaluates to numeric and the target database file is not INDEXed, then the target record pointer moves to the record number returned by the linking expression each time the source database file pointer moves.

   For example, the following linking expression relates two unINDEXed database files so each pair of adjacent records in the source database file points to a single record in the target file.

   ```
   IIF(MOD(RECNO( ),2)=1,INT(RECNO( )/2)+1,INT(RECNO( )/2))
   ```

   Record numbers one and two in the source point to record number one in the target, three and four in the source point to two in the target, and so on.

   This type of relation can be used to build virtual data structures based on a link that is a combination of value and physical location.

In general, it is important to your mastery of dBASE to understand the truly dynamic nature of relating database files with SET RELATION TO.

For more information on SET RELATION TO, refer to the page D3-11 of the April 1985 issue of ***TechNotes***.

INKEY( )

1. The INKEY( ) function in the Developer's Release and dBASE III PLUS seems to return the wrong answer after a WAIT command has been executed. For example,

```
WAIT
i = INKEY( )
? i
```

   The memory variable "i" will always contain the value zero no matter what key was last pressed. The reason for this is that WAIT pauses and waits for a key value from the keyboard or type-ahead buffer. Once the key is sensed, it is taken from the buffer. The INKEY( ) function also looks at this buffer for a key, but finds that the last key has been taken by the WAIT command and so returns a zero value.

   The correct method to poll the keyboard for a key press is to use either the WAIT command or the INKEY( ) function but not both in combination. For example,

```
* —-Use the WAIT command.
WAIT TO choice
DO CASE
CASE choice = 19
        ...
CASE choice = 27
        ...
ENDCASE
*
* —-Use the INKEY( ) function.
key = INKEY( )
DO WHILE key = 0
    key = INKEY( )
ENDDO
DO CASE
CASE key = 19
        ...
CASE key = 27
        ...
ENDCASE
```

2. The following program can be used to help you determine the key code for each key pressed.

```
SET ESCAPE OFF
SET TALK OFF
esc = 27
DO WHILE .T.
    key = INKEY( )
    DO WHILE key = 0
        key = INKEY( )
    ENDDO
    ? key
    IF key = esc
       EXIT
    ENDIF
ENDDO
SET TALK ON
SET ESCAPE ON
RETURN
```

Framework II

by Chris Lo Vasco

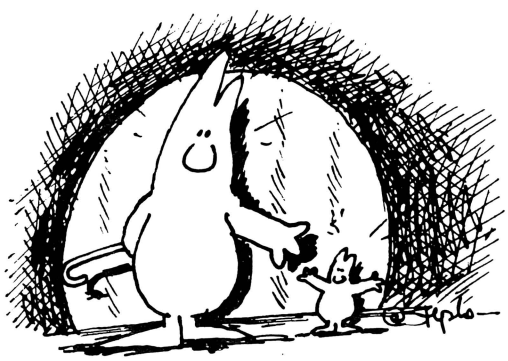
Macro for Superscript and Subscript

Many users have asked if there is an easy way to change the print style of selected characters to either superscript or subscript. The basic concept is presented in "Framework II Printer Drivers" in this issue, but if you prefer to select options from a menu (and not have to remember which ASCII code does what), the following macro can be placed in your Library and invoked at any time.

First, create an empty word frame and label it {Alt-<key>} where the <key> is any valid macro key that has not already been assigned. I chose {Alt-Z} for mine. Inside this frame create four empty word frames. Label them as follows:

Frame #	Label
1	1-Superscript [ON]
2	2-Superscript [OFF]
3	3-Subscript [ON]
4	4-Subscript [OFF]

You may want each menu choice to be accompanied by a description, centered at the bottom of the screen. **Dnlevel** into each frame and enter the desired text.



Now highlight frame number 1 (labeled 1-Superscript [ON]).
Press **EDIT FORMULA (F2)** and enter the following formula:

@performkeys(@chr(244)),@quitmenu

Press **Uplevel** and repeat this for frames 2,3, and 4 using **@chr**(245), **@chr**(246), and **@chr**(247) respectively.

Return to highlight the containing frame ({Alt-Z} in my example) and press **F2.** Enter this formula:

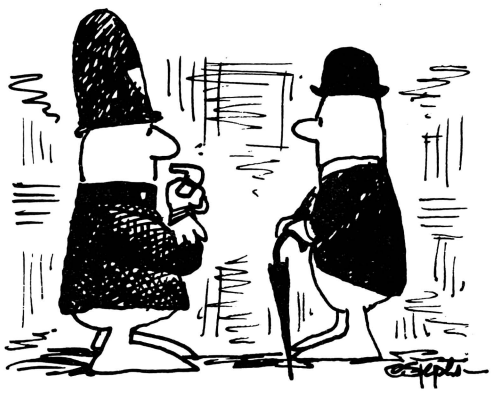
@menu([{Alt-Z}])

Press UPLEVEL. Now you may MOVE **(F7)** the containing frame to the Library section labeled Macros. Whenever you need superscript or subscript, press **Alt-Z** and make your selection. You can do this by highlighting your selection (use the cursor keys) or pressing a number from one to four.

Comparing English Date Strings

Readers of ***TechNotes*** may recall an article in the November 1985 issue that dealt with date and time calculations in Framework and Framework II. The date format used then was mm/dd/yy. But what about using the date format generated by Framework II's **@date1** and **@date4** functions? For example, the formula **@date1**(**@date**(1986,6,12) returns the English language result: Jun 12, 1986. Here, then, is the formula that will enable you to compare English dates.

To make the formula accessible from any spreadsheet frame, database frame, or other FRED programs, place it in the formula area of an empty word frame called EnglDate, and place Engldate in the Library cabinet.



```
; EnglDate
; by Jack Hirsch
@diffdate(
     @today,
     @date(
          @value(@mid(@item1,@len(@item1)-3,4)),
          @ceiling(@scan(
               @mid(@item1,1,3),
               "JanFebMarAprMayJunJulAugSepOctNovDec")/3
          ),
          @value(@mid(
               @item1,
               @scan(" ",@item1)+1,
               @scan(",",@item1)-@scan(" ",@item1)-1)
          )
     )
)
```

For this example, create a database frame. One field is labeled Call Date, another is Days Since Call. Call Date has dates that have been entered in the full English format described above, i.e., Jun 3, 1986. In the formula area of Days Since Call place the following formula:

Days Since Call := @EnglDate(Call Date)

When Days Since Call is recalculated, the result will be the number of days that have elapsed since Call Date.

In addition to providing useful information, Days Since Call can be used for either an ascending or descending sort, so now you can sort your data by date even though the dates are entered in a character format.

The formula can be used in this fashion from a spreadsheet frame as well.

Macro to Backup Files From Hard Disk to Floppy

There are times when it would be extremely convenient to be able to copy a block of files directly from a file cabinet to a floppy disk drive. However, Framework II requires that these files be loaded to the desktop before copying them over to a floppy. In an effort to overcome this, the following macro was written to utilize DOS Access to accomplish this copying of files.

In order to use this macro, you must first create an empty word frame in the Macros section of the Library. Name this frame using the **Alt**-keystroke sequence you want to invoke it. We'll use {Alt-B} for this example. Inside the {Alt-B} frame create another empty word frame and title it Copyprog.

Put the following formula into the formula section of the {Alt-B} frame:

```
@prompt("Highlight the files and press return"),
@keyfilter({Return},[{Alt-B}].Copyprog)
```

Place the following formula into the formula section of the Copyprog frame:

```
@local(cnt,files),
files:=@panel1,
@performkeys("{Esc}"),
files:=files-@panel1+1,
@performkeys("{Ctrl-Uplevel}{Ctrl-C}EFwbackup"&
     "{Return}{F4}{F4}{Return}{Scroll-Lock}{Ctrl-Dnlevel}"),
cnt:=1,
Fwbackup:="",
@while(cnt<=files,
     Fwbackup:=Fwbackup&"copy " & @setselection & " A:",
     @performkeys("{Dnarrow}"),
     cnt:=cnt+1
),
@keyfilter({All}),
@writetextfile("fwbackup.bat",Fwbackup),
@performkeys("{Ctrl-Uplevel}{Ctrl-C}EResults" &
     "{Return}{F4}{F4}{Return}"),
@run(Results,"\COMMAND.COM /C FWBACKUP.BAT"),
@setselection("Fwbackup"),
@performkeys("{Del}")
```

This macro can be invoked from anywhere on the desktop. It is written to copy the files to the A: drive. After it is invoked, and when prompted, move to the proper file cabinet and highlight the files you want to be copied. If you highlight a subdirectory name, the whole subdirectory will be backed up. You must highlight a block of files, moving from top to bottom. When you have selected the files and/or directories you want to backup, press **Return**. The macro will take over from here and perform the following steps:

1. Create two frames on the desktop (Results and Fwbackup).
2. Process the highlighted file names and write the Fwbackup frame out as a batch file.
3. Execute the batch file.
4. Display the results in the Results frame.

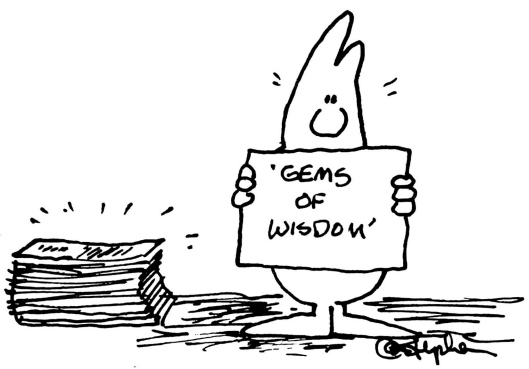
An important tip from the {Alt-B} macro is that **@setselection** can be used to retrieve the filename of a highlighted file in a drive cabinet.

Sheet Feeder Print Macro

Here is a short macro for users who have sheet feeders on their printers and must turn the **Print:Wait for Each Page** on before printing. The frame is named {Alt-P} and contains the following formula:

```
@performkeys(''{Ctrl-P}''),
@if(@sense(#PW),@performkeys(''b''),@performkeys(''wb''))
```

When you press **Alt-P**, the macro will make sure that **Wait for Each Page** is ON.



CORRIGENDA

On page 40 of the June 1986 issue of ***TechNotes***, the second example of **@writetextfile** in the Usage Tips is incorrect. It should read:

```
@local (str),
str :=''@writetextfile('' '' '' & @setdrive & @setdirectory &
      ''\TEXT.TXT'' '',FRAME)'',
@str
```

ON-LINE SERVICES

Do you have a modem? Do you subscribe to CompuServe, The Source, or Onepoint? If so, we have great news for you! Ashton-Tate offers technical support on all of these electronic information services. You can quickly access the kind of quality technical information that you have come to expect from ***TechNotes***, and you can do it through your computer system. To take advantage of this progressive new support method, you must subscribe to one of the services listed below. There are no additional charges for using our on-line support services, apart from the normal connect-time fees for each service. You can get specific subscription information by calling one of the following numbers:

Service	Subscriber Information	Command to find Ashton-Tate
CompuServe	(800) 848-8199	GO ASHTON
The Source	(800) 336-3366	ATSIG
Onepoint	(800) 222-2250	

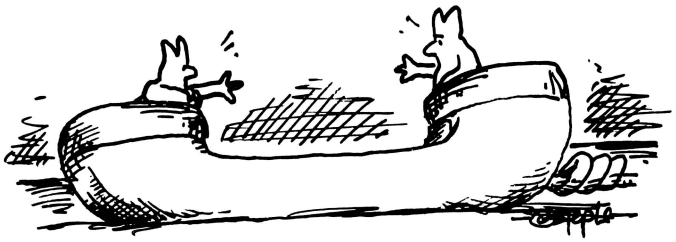
To give you an idea of the fast-moving dynamic support that is happening on-line, ***TechNotes*** will be publishing regular excerpts from the Ashton-Tate Forum on CompuServe. This monthly material comes from messages our Forum users have posted on the bulletin board. To retain the original sense of this material, we have edited only for spelling, and all message headers have been removed to protect the identities of our users. Every other message within a single conversation topic has been italicized to enhance readability.

---

dBASE II, dBASE III, dBASE III PLUS:

I ran into an unusual problem yesterday while attempting to use the RUN command out of dBASE III PLUS. I had written two small utility programs for a client...The first was a disk format utility (! FORMAT A:), and the second a DOS backup (! BACKUP \DBASE\CLIENT*.DBF A:). The FORMAT and BACKUP files reside in a subdirectory, DOS, which is specified in his path. Upon RUNning the format utility, and after inserting a blank floppy in the A drive, I received an error message that the diskette could not be formatted (error track 0, if I remember correctly). The same command run from the DOS prompt (Same directory \DBASE\CLIENT), with the SAME diskette worked fine. Attempting to RUN the backup program from within dBASE produced an invalid drive specification (A drive). Again, the backup worked just fine from the system prompt. Does anyone have an idea as to what the problem may be? (by the way.....other RUN commands such as !ERASE TEMP.DBF, work just fine).

One of my clients just found the same problem yesterday. Even though the ! BACKUP ..... works fine on my machine, it won't on theirs. When you say batch files, you mean from within dBASE with !, right ?



I had some problems with the BACKUP command in v. 1.0. Actually it worked, or appeared to, but had trouble restoring except from dBASE. Weird. Try a batch file (RUN <batname>) or better yet, try RUN COMMAND /C BACKUP.... Should work much better.

This is another case of the DMA boundary error problem that has been well known (and groaned about) on the IBM Forum. Apparently, depending on what is allocated in your memory (more specifically, where your first page of "free" memory to RUN from the dot prompt is), you may get funny behavior with FORMAT, DISKCOPY, BACKUP, etc.). It isn't (fortunately) a dBASE anomaly; just by using certain combinations of resident programs, BUFFERS= and FILES= statements, you'll see the problem from DOS itself.

Try changing your BUFFERS or FILES statement, and try the offending sequence again. I'll bet you find it works OK then! This seems to be related to the DOS environment space.

.....

I am thinking of switching from a Northstar Horizon CP/M to a MS-DOS machine, such as a Compaq or IBM or compatible. I have done much development in dBASE II on this CP/M machine. I am wondering what the chances are of making the switch to dBASE III PLUS, and bringing along the dBASE II command files and file structures along with the databases I've developed.?

You shouldn't have too much trouble getting things running. Just familiarize yourself with the subtle changes; $/SUBSTR( ), EOF/EOF( ), SELECT PRIMARY/SELECT <alias>, etc. I'd recommend that you copy your DBF files out SDF and handle the conversion of data that way.

In dBASE III (including PLUS), there is a supplement to the manual (included) that details the changes between dBASE II and that particular version of III: "dBASE Bridge", originally written by Adam Greene and later edited to cover dBASE III PLUS. All dBASE II files can be transferred AS IS to a DOS machine (either via RS-232, modem, or one of the utility programs like Uniform, Media Master, Xeno-Copy, etc.). Rename the CMD files to .PRG files, then either use them with a dBASE II on the DOS machine, or use the DCONVERT.EXE program (supplied with all versions of dBASE III) to convert all of your files over to dBASE III.

.....

Can anyone explain what the "R" function in a PICTURE specification does? The description in the dBASE III manual is cryptic, I see no examples.

@R PICTURE function allows the embedding of display characters in the field. i.e., GET PHONE PICT "@R (999)999-9999".

I think I understand, although it sounds strange. Are you saying that if you specify @R you can include literal characters in the template, and if you don't, you can't? It sounds strange because it's a halfway feature. You presumably can't include any character that has meaning in a template, since there's no way to indicate which characters are literal and which are not. It's also an unnecessary complication—as long as you're limited to using characters that have no meaning in the template, there's no logical reason to have to enter @R at all. Right? Or do I have a mistaken impression?

Literal characters in the template without the @R will get characters in a weird way. Maybe there's supposed to be a use for that. Example: GET phone PICT "(999)999-9999" will result in phone = "(123)456-7890" where GET phone PICT "@R (999)999-9999" results in phone = "1234567890". If phone is not long enough in the first case, results will be truncated (on the right I believe).

You could have a field stored as 9999999999, and display or print it as 999-999-9999 by using the @R function. This would be helpful if you desired not to store the dashes in your field.

You should add that it's really only of benefit with the SAY command, as in @ rw,cl SAY "9999999999" PICTURE "@R 999-999-9999". Serves no value with GETS.

Framework II:

I'm going crazy trying to work around the fact @setselection will accept only a "string" argument in the version of Framework II that I am using. I want to use Darryl Rubin's Glossmac [Published in *Ashton-Tate Quarterly,* second quarter '86 —Ed.]. How do I obtain a version of Framework II that allows variables with @setselection?

@setselection does, indeed, accept variable names. Try the following:

```
@local(frame),
@eraseprompt,@beep,
frame := @inputline("Enter name of frame to go to"),
@setselection(frame)
```

And supply the name of a frame that exists on the desktop. I tried Glossmac with my Framework II and it worked fine. Hope this is of help.

.....

I own a Prowriter 8510 and am using it with Framework II. I am unable to get it to single space printouts even when I have the single space option specified. Double spacing gives me quadruple spaced print. I guess you get the idea. Can you give me any help on how I might be able to correct this situation, if it can be corrected. Much obliged!

Check SW-1 position 8. It is the function of CR. If the switch is closed a Carriage return also generates a line feed. The switch should be open toward the front of the printer, so that extra line feeds are not generated by the printer itself. This may be your problem.

I suspect that your printer is set for "Auto LF after CR". This is an option that can be set by DIP switches on some printers, and by control codes on others. Framework II requires that no Line Feed be sent after the Carriage Return (also that NO skip-over-perforation is occurring).

Once again, thank you. I was able to overcome my problem by using the switches as we previously discussed. I had tried the settings once before and it didn't seem like they had taken effect but when I tried again everything worked fine. I appreciate the advice and look forward to a lot of happy hours with Framework.

.....

Can anybody tell me if there is an upgrade available for TimeFrame? My copy does not seem to work properly with Framework II. I cannot find any discussion on this problem so it might just be my copy? Please advise, as I can not even get into the tutorial for the calendar/appointments section.

There certainly is! Take the original TimeFrame diskette and send it to Ashton-Tate Publications, Attn: Maxine Harris, 20101 Hamilton Avenue, Torrance CA 90502-1319. Attach a brief note with a return shipping address and why you need the update (just to keep things clear), and Maxine will send you a new disk that will work with both versions of Framework (1.1 and II). Keep your documentation, as that is valid for both the old and new versions; supplemental information that explains the differences between the two will come with the new disk.

.....

MULTIMATE TECHNICAL SUPPORT

by Liz Burke

Common Printer Questions

Characters in Left Margin

WHAT IF... when printing a document, characters are printing out in the left margin?

Are you specifying the correct PAT on the printer parameter screen? You can translate the codes printing in the left margin into hex and then check the PAT to see if they are coming from the PAT. If codes are coming from the PAT, check for printer dipswitches that pertain to "Allow Printer to Receive Escape Sequences from Host (PC)." If your printer has one, set the switch to on. If characters are only printing on the first line, try putting a **Return** at the top of the page.

Creeping Text

WHAT IF... you are printing with "Number of Original Copies" more than one and creep occurs on the second and third copies?

Place the "Printer Reset" code in "Printer Initialization" and in "Printer Reinitialization," if it is not already there. This will ensure that the printer resets before and after each new copy. Also, place the "Set Page Length in Inches" code into the "Set Page Length in Inches" area of the PAT (Printer Action Table), and OC in formfeed.

Document Flashing in Queue

WHAT IF... you were printing with no problems and now your printer will not print. Whenever you try to print, the name of your document ends up flashing in the Queue?

Verify that the print parameters are set up correctly. For version 3.22, check the print parameters in the "Submit a Document for Printing" screen. For version 3.30 or higher, check the print parameters in the "Print Parameters" screen. Also, be sure the correct PAT file is on the MultiMate Boot Disk, and is spelled correctly (with no leading spaces) in the print parameters screen. Lastly, check the drive defaults to make sure they are accurate for your computer.

Document Held in Queue

WHAT IF... your document is not being released from the Queue after ERASEing WPQUE.SYS?

Try creating a CONFIG.SYS file with FILES=15 and reboot, or put MODE LPT1:,,P in your AUTOEXEC.BAT file and reboot.

Document in Queue

WHAT IF... you submit a document to print, your printer does not print, your document shows up in the Queue Control, and you get the error message "Document is Currently Printing" when you try to edit the document?

Return to DOS and delete the file WPQUE.SYS from your MultiMate directory with the command line: ERASE WPQUE.SYS.

Formatting Problems

WHAT IF... you are having a problem printing out the first line of a document and the centering or left margins are not accurate, or superscripts and subscripts are printing a full line up or down?

First, check to see if there is a **Return** at the top of the page. If not, put one in and try printing again. The reason for putting in the **Return** is to bring the printer carriage back to the left margin. The "Printer Reset" codes in "Initialization" may be ignored. Also, make sure the headers are set up correctly. If there are any control codes or pitch changes on the first line, try inserting a **Return** as well.

Keyboard Locking up

WHAT IF... you use MultiMate Advantage, and when you press **Ctrl-Break**, keyboard locks up?

Try pressing **Ctrl-Break** again.

Line Spacing

WHAT IF... your document is printing double spaced but your format line is set to single spacing?

1. Does a print screen generate double line spacing?

   Check your printer's dipswitches and set the line feed switch off. Reset your printer and try again.

2. Are you working with the correct PAT?

   If yes, check with MultiMate Technical Support for the latest PAT that may have new linespacing codes.

3. What kind of computer are you using?

   If it is a Tandy, delete line feed codes from the PAT. The Tandy PC automatically issues a line feed.

Printing

WHAT IF... your printer won't print?

Check to see if the printer is plugged in and the cable is connected securely. Can you do a print screen? The following conditions can be used if you cannot get a printout.

```
COPY CON LPT1, LPT2, LPT3
this is LPT1
Ctrl-Z
```

OR

COPY CON COM1, COM2
this is COM1
Ctrl-Z

If you get the message "1 file(s) copied" and then nothing happens, either the port does not exist or nothing is hooked up. If the message "1 file(s) copied" does not appear, nothing prints, and system locks up or "error writing device" message appears, a device is hooked up but there is a problem communicating to it. Check to see if the printer is out of paper, the printer is off-line, or any other channel is blocked.

Print Error Accessing

WHAT IF... you submit a document to print and you get the error message "Print Error Accessing TTYCRLF.PAT"?

Make sure the printer defaults are correct: spelling of the PAT, drive defaults, correct PAT being used, and PAT is on the working MultiMate Boot Disk. If everything appears correct, replace the TTYCRLF.PAT file with the PAT that suits your printer (i.e., EPSONFX or DIABL630) on the "Print Parameters Screen" instead of using the all-purpose PAT, TTYCRLF.PAT. Make sure the replacement PAT file is on the working MultiMate Boot Disk.

Print Error Accessing

WHAT IF... at the MultiMate Main Menu you get the error message "Print Error Accessing @: Press Escape To Continue," and the computer freezes?

Reboot your computer, and at the DOS prompt where MulitMate is located, type ERASE WPQUE.SYS.

Printer Needs Attention

WHAT IF... every time you send a document to print, you get the message "Printer Needs Attention, Press Escape to Continue" after printing a paragraph? If you press **Esc**, another paragraph prints and you get the error message again.

Edit the PAT and change "Seconds for Printer Timeout" (ver. 3.30 or higher) or "Number of Seconds to Wait before Reporting an Error" (ver 3.22 or lower) to 90.

3.30 or higher (Enter code in decimal)
3.22 or lower (Enter code in hex)

MultiMate is checking the printer too soon to see if it is ready to receive information. Extending this number will remedy the problem. Another reason for this error message is that MultiMate checks for printer status errors before printing to the printer. Resubmit your document to print and on the "Print Parameters Screen" specify "L" instead of "P." If you have a serial printer, specify "A" instead of "S" to print. This will instruct MultiMate to print through DOS instead of through the MultiMate print routine. You may have more success printing this way since DOS does not check for printer status error messages from the printer before printing.

Proportional Spacing

WHAT IF... you enter "Y" to proportional spacing, "M" for micro on the MultiMate Print Screen, and when you print, the centering, indents, and tabs do not line up?

Check to see if you are using a proportionally spaced print wheel. If you are using a proportionally spaced print wheel, replace it with a 10 CPI print wheel. MultiMate does proportional spacing based on 10 CPI and requires a 10 pitch wheel in the printer. To use a 12 CPI print wheel with version 3.30 and higher of Multimate, you need to create a new CWT (Character Width Translation) table that has the widths of each character on the 12 CPI wheel.

Stop Print

WHAT IF... you submit a document to print and you get the error message "Requested Stop Print. Press Escape To Continue?"

The print program has encountered the stop print function you have placed in your document with **Alt-P**. To continue printing, press **Esc**. If you have a printer buffer, this message may appear on the screen but your printer will still be printing.

MultiMate Magic

Inserting a Page

WHAT IF... you need to insert an additional page of text in between pages three and four of a five-page document?

Put the cursor at the bottom of page 3 and press **F2**, Page Break. A new blank page four will appear on screen. The original pages four and five will now become pages five and six.

Library Attachment

WHAT IF... you try to attach a Library entry to a document, the Library tab settings are ignored, and the information is lined up with the tab settings in the document? How can you bring forward the same format line on the Library entry page?

To have Library entries include the format line you wish, insert an additional format line at the top of the entry page using **Alt-F9**. This format line will then be brought forward when you access the Library entry.

Line and Box Drawing

WHAT IF... you want to insert line and box drawing characters into a document with a key procedure?

Insert a page break into your document by pressing **F2** on the first character of the line that will follow the key procedure. Use **PgUp** to return to the bottom of the previous page. Execute the key procedure and then combine the two pages using **Shift-F2** or repaginate your document using **Ctrl-F2**.

Selective Merging

WHAT IF... you want to merge a form letter with only part of your mailing list?

You can accomplish this task easily with MultiMate version 3.22 or earlier by creating a new merge data file and then using the External Copy function, **Shift-F8**, to copy the names and addresses to send into a new file. You then merge your letter with this new file. With version 3.30 and higher, you can instruct MultiMate to select the block of records you want to merge print.

INTERNATIONAL SUPPORT

by Paul Bukauskas

Reassigning Keyboard Characters in Framework

Submitted by Ernesto Mendez Solis

A commonly-asked question of our Mexican users is whether or not Framework II will support foreign character sets. Foreign translations of Framework II fully support foreign keyboards and DOS. However, in those cases where only an American keyboard is available, the keyboard may be redefined using FRED macros. That is, the keyboard character "A" may be assigned the alternate value "á" when **Alt-A** is used.

Redefining the keyboard is no more difficult than consulting an ASCII table. For example, the following letters could be added to the keyboard:

Character	ASCII value
á	160
é	130
í	161
ó	162
ú	163
ñ	164
Ñ	165
¿	168
¡	173

To produce one of these characters, press the **Alt** key while typing in the ASCII value from the numeric keypad. Release the **Alt** key after all three ASCII digits have been entered.

By creating a FRED macro you need not remember the individual ASCII values. Create a macro in the following way:

1. Pull down the Create Menu and choose the **Macro/Abbreviation** option: **Ctrl-C M**.
2. Press the **Alt** key and the letter you wish to program; For example, **Alt-A**.
3. While pressing the **Alt** key, type 160 from the numeric keypad.
4. Finish the recording session by pressing **Ctrl-Break**.

   Now, each time you press **Alt-A** the letter "á" will appear on the screen. Remember, these macros are automatically placed in the Library, but not saved until the Library is saved.

Ashton-Tate in Asia

Ashton-Tate World Trade Corporation is an Ashton-Tate subsidiary that handles technical support for Japan, Hong Kong, Singapore, Malaysia, Thailand, India, Pakistan, Taiwan (R.O.C.), China (P.R.C.), Korea, Philippines, Brunei, and Indonesia. Nippon Ashton-Tate is a joint venture of Ashton-Tate and Japan System Engineering (JSE). Nippon Ashton-Tate distributes English and Japanese (Hirajana, Kanji, and Katakana) versions of Ashton-Tate products in Japan.

The latest release of dBASE II, version 2.0J, is available for the following computers:

Computers	Operating Systems
Canon AS Series	MS-DOS
Fujitsu FM Series	MS-DOS, CP/M-86
Hitachi B16 Series	MS-DOS
IBM 5550 Series	MS-DOS
NEC 5200 Series	MS-DOS
NEC PC-9801 Series	MS-DOS, CP/M-86
Mitsubishi Multi 16 III	CONCURRENT CP/M
Mitsubishi Multi Series	CP/M-86
OKI if800 Series	MS-DOS
Sanyo MBC 6800	CONCURRENT CP/M
Toshiba Pasopia Series	MS-DOS

The most current release of dBASE III is 3.0J, available for the following MS-DOS machines: NEC PC-9801 Series, IBM 5550 Series, and the Fujitsu FM-10 Series.

Nippon Ashton-Tate offers Japan English-speaking sales and support for dBASE III PLUS, Framework II, MultiMate, and Javelin. Those in other Asian countries should contact a local dealer. For the location of the nearest dealer, contact Ashton-Tate World Trade:

Ashton-Tate World Trade Corp
Aoyama Success Bldg
2-11-13 Minami Aoyama
Minato-ku, Tokyo, Japan 107
Phone : (03) 479-7191
Telex : 02422532 ATJAPNJ

Nippon Ashton-Tate
Minato-ku, Tokyo, Japan 107
Phone : (03) 479-7621
Fax : (03) 479-7628

LETTERS

Phones Busy - Long Hold Time

Dear ***TechNotes***:

Over the past several months, I have tried to call the Support Center. Many times, I will get a busy signal. When I finally do get through, the wait time has been 10 minutes. When is the best time to call the Support Center?

R.W.

Dear R.W.:

Since the simultaneous release of dBASE III PLUS and Framework II, the number of incoming calls increased sharply. This resulted in a long hold time, where it was possible to wait up to 20 minutes for a technician. Many customers complained about this excessive hold time. Since dBASE III PLUS and Framework II sales have continued to rise, the Support Center has hired several more technicians. During the month of June alone, eight new technicians were added. Operators have also been hired to screen the call to make sure your software is registered with us. In addition, the manager of the Support Center has limited the number of incoming lines, so that the wait time for a technician is reduced. That is the reason for the busy signal. It is difficult to suggest the best time to call the Support Center. However, our busiest hours are usually between 11:00 am and 2:00 pm, Pacific Time.

Changing Debit and Credit Signs

Dear ***TechNotes***:

The instructions on page U5-25, lines 6 and 7 of the *dBASE III PLUS* manual describes the functions that display "CR (credit) after a positive number" and "DB (debit) after a negative number." This is contrary to general bookkeeping practice. On the other hand, page P7-5 line 5, describes correctly these functions. However, the function operates incorrectly by applying CR to positive numbers and DB to negative numbers. A correction can be made by patching DBASE.MSG as follows:

Assuming MS-DOS utility DEBUG.COM and DBASE.MSG are located on the same drive and subdirectory (user input is in italics),

C>*DEBUG DBASE.MSG* **Return**
-E16EE **Return**
xxxx:16EE 44.*43* **Spacebar** 42.*52* **Return**
-E16F2 **Return**
xxxx:16F2 43.*44* **Spacebar** 52.*42* **Return**
-W **Return**
Writing 2FF4 bytes
-Q **Return**
C>

The "@XC" function works fine as long as you use them together. If one is used without the other, their meanings are reversed.

W.S.B.

Dear W.S.B.:

Thank you for passing along the information.

Zip Codes with the Letter l instead of the Digit 1

Dear ***TechNotes***:

While entering zip codes into a dBASE III PLUS database file, I accidentally used a lowercase l instead of the digit 1. The database file has several thousand records. Is there a way to change the l's to 1's?

S.P.

Dear S.P.:

By using the AT( ) function in conjunction with the STUFF( ) function, it is possible to REPLACE all l's with 1's.

The AT(x,y) function returns the position where the string "x" occurs in string "y." If the string "x" does not occur in the string "y," the AT(x,y) function will return a zero. For example, suppose we are searching for the string "DAYS" in "HAPPY DAYS." AT("DAYS","HAPPY DAYS") returns the number seven, which is where "DAYS" begins in "HAPPY DAYS."

The STUFF( ) function allows the user to insert a string inside an already existing string. The syntax is:

STUFF(x,a,b,y)

The string "y" is to be inserted into string "x" starting at position "a" and replacing the next "b" characters. For example:

? STUFF("ABXYE",3,2,"CD")

results in "ABCDE."

The string "CD" is to be inserted into "ABXYE," starting at the third position (where the "X" is positioned) and replace the next two characters ("XY"). In essence, we are replacing the "XY" in "ABXYE" with "CD," so that it results in "ABCDE."

We can now design a program which uses these two functions. Assuming that Zip is the zip code field in the database file Names.DBF:

```
USE Names
DO WHILE .NOT. EOF( )
    * —-Check to see if an "l" is in the Zip field.
    DO WHILE AT("l",Zip) > 0
        REPLACE Zip WITH STUFF(Zip,AT("l",Zip),1,"1")
    ENDDO
    SKIP
ENDDO
```

Notice that the AT( ) function is also used inside the Zip field. The AT("l",Zip) returns the start position of the string that is to be replaced. The rest should be explanatory from above.

If you know there is not more than one l in each Zip code, we could easily say:

```
REPLACE ALL Zip WITH STUFF(Zip,AT("l",Zip),1,"1")
```

However, to make sure that all l's are replaced, the DO WHILE AT("l",Zip) > 0 is used. Once one l is replaced, it checks again to see if another l exists.

Totals in REPORT FORM with Character Output

Dear ***TechNotes***:

In the March 1986 issue of ***TechNotes***, there was a discussion of the IIF( ) function by Christopher White. He mentioned that this function could be used to suppress zeroes in numeric fields with a value of zero. However, the suggested expression to be inserted into a report form:

```
IIF(Amount=0, "     -", STR(Amount,6))
```

defeats the major points of having a numeric field in reports which is, of course, to be able to total the field. Is there any way to suppress zeroes and at the same time, be able to total the field?

R.C.S.

Dear R.C.S.

No, there is not. This dilemma also occurs when you use the TRANSFORM( ) function to insert commas in numeric output. However, if you are not doing subtotals, but want totals, MODIFY the REPORT and specify no totals. Use the REPORT FORM with the NOEJECT clause. This will stop the REPORT FORM on the last line. Use the SUM command to sum the fields to memory variables. Then use the ? command with SET PRINT ON and output the totals on the next line in their appropriate column. For example, suppose Names.DBF contains two fields, Name and Amount. I CREATE a REPORT Rpt.FRM that contains Name in one column and IIF(Amount=0,"   -",STR(Amount,6)) in the next column. The following program will give the desired result.

```
REPORT FORM Rpt TO PRINT NOEJECT
SUM Amount TO mamount
SET PRINT ON
* —-Substitute x below to give correct positioning.
? "Total" + SPACE(x) + STR(Amount,6)
SET PRINT OFF
EJECT
```

TechNotes Documentation Survey

Ashton-Tate wants to provide you with product documentation that meets your learning needs. The following questions are designed to help us gather information about the types of documentation Framework II users find most helpful. Your answers will be used to assist our Documentation department in improving the effectiveness of our product manuals.

The first part of the questionnaire requests information about you and your computer system, because documentation that would be most useful to one user may not be most useful to another. The second part of the questionnaire requests information about your reactions to the Framework II documentation. When you have completed the questionnaire, use the enclosed postage-paid envelope to return it to us. Thank you for your time and feedback.

1. What is your personal computer experience?
   ___ Novice personal computer user
   ___ Intermediate personal computer user
   ___ Advanced personal computer user, non-programmer
   ___ Advanced personal computer user, programmer

2. On what type of computer system are you using the product:
   a. Drives:
   ___ PC with 2 floppy disk drives
   ___ PC with a hard disk drive
   ___ Part of a network of PCs
   b. Monitor: ___ Monochrome ___ Color
   c. Internal memory (RAM):
   ___ 128K ___ 256K ___ 384K ___ 512K ___ 640K ___ more than 640K
   d. Hard disk storage capacity:
   ___ 10M ___ 20M ___ 30M to 40M ___ more than 40M

3. Did you notice that there was an on-disk tutorial? ___ Yes ___ No

4. Did you use the on-disk tutorial to learn Framework II? ___ Yes ___ No

5. Was the on-disk tutorial adequate to get you started? ___ Yes ___ No

6. Did you use the printed tutorial, *Learning Framework II*, to learn the program?
   ___ Yes ___ No

7. Is it a good idea to expand the Help system to include more of the documentation?
   ___ Yes ___ No

8. Which is your one, most preferred way to learn Framework II?
   ___ Printed manual ___ Videotape
   ___ On-disk tutorial ___ Laserdisk
   ___ Audiotape ___ Classroom instruction
   ___ Work with user group
   ___ Looking at someone else's frames, work, or FRED source code
   ___ Other _____

9. How often do you use the HELP feature in the Framework II software?
   ___ Often ___ Sometimes ___ Rarely ___ Never

10. How useful do you find the HELP feature in the Framework II software?
    ___ Excellent ___ Good ___ Neutral ___ Fair ___ Poor

11. What one thing would most improve the Help feature?

_____

12. How do you rate the Framework II documentation?

	Excellent	Good	Neutral	Fair	Poor
Ease of learning	___	___	___	___	___
Clarity of information	___	___	___	___	___
Sufficient information	___	___	___	___	___
Usefulness of examples	___	___	___	___	___
Quantity of examples	___	___	___	___	___
Organization of information	___	___	___	___	___
Usefulness of illustrations	___	___	___	___	___
Attractiveness of manuals	___	___	___	___	___

13. How often do you use the following parts of the Framework II documentation:

	Often	Sometimes	Rarely	Never
Getting Started	___	___	___	___
Learning Framework II	___	___	___	___
Using Framework II	___	___	___	___
Framework II Advanced Topics	___	___	___	___

14. Did you know that Framework II has its own programming language, called FRED? ___ Yes ___ No

15. How many FRED programs have you developed?
___ None ___ 1-10 ___ 11-25 ___ 26-50 ___ more than 50

16. How useful was the documentation in teaching you to build FRED language programs? ___ Excellent ___ Good ___ Neutral ___ Fair ___ Poor

17. What one thing would improve the documentation's presentation of the FRED programming language?

_____

18. Do you use Framework II for telecommunications? ___ Yes ___ No

19. How useful was the documentation in teaching you to use Framework's telecommunications features?
___ Excellent ___ Good ___ Neutral ___ Fair ___ Poor

20. What one thing would improve the documentation's presentation of Framework II telecommunications?

_____

21. What is your single, most important Framework II learning need that is not currently being met?

_____

22. Does your organization provide training in how to use Framework II?
___ Yes ___ No

23. If your organization provides Framework II training, what learning needs were met that were not satisfied by the documentation?

_____

24. Are you interested in participating in further research into the learning needs of Framework users? ___ Yes ___ No

If yes, please provide your name, address, and phone number below.

_____

_____

_____

SUBSCRIBE TODAY TO *TechNotes*!

Don't miss a single issue of Ashton-Tate's hottest source of technical know-how! Each monthly issue of ***TechNotes*** provides you with the most current, accurate technical information available from the Software Support Center on dBASE II, dBASE III PLUS, and Framework II.

Call our toll free number to subscribe:

(800) 331-4164 (In California)
(800) 828-2514 (Residents of other states)
(619) 487-3212 (In Canada)

If you prefer, fill out the form below and return it with payment to:

***TechNotes* Subscription Service**
P.O. Box 3729
Escondido, CA 92025-0929

— — — — — — — — — — — — — **(CUT HERE)** — — — — — — — — — — — — —

NOW ACCEPTING CREDIT ORDERS! Call our toll free number for details. Make checks payable to ASHTON-TATE.

Amount Enclosed:

__ $50.00
__ $65.00 (Extra Postage Canada and Mexico)
__ $75.00 (Extra Postage other foreign countries)
__ Bill Me

Form of Payment: (check one)

__ Check (**U.S. dollars drawn on U.S. bank**)
__ VISA Number _____
__ MasterCard Number _____
Expiration Date _____
Signature _____

Check One: __ New Subscription __ Renewal

Name _____

Company _____

Address _____

City _____ **State/Province** _____

Zip/Postal Code _____ **Country** _____

Which product(s) do you use? __ **dBASE II** (D2)
__ **dBASE III** (D3)
__ **Framework** (FW)

Please allow 30 to 60 days for delivery of your first issue. Ashton-Tate's Federal Tax Identification Number is 95-3547543.

1000

TechNotes
BACK ISSUES ORDER FORM

Maybe your subscription to ***TechNotes*** just began and you want to have every issue. Maybe you don't subscribe, but want to receive last month's issue. Maybe you want to refer to an article in a past issue to help you design that special application.

Whatever the reason, order Back Issues! Add to your library of dBASE II, dBASE III PLUS , and Framework II technical reference material. To order your Back Issue(s), fill out this form, enclose your payment, and return to:

***TechNotes* Back Issues**
Ashton-Tate Software Support Center
Attn: Kathy Smith
20101 Hamilton Ave.
Torrance, CA 90502-1319

Name _____

Company _____

Address _____

City _____ State/Province _____ Zip Code _____

Country _____ Phone Number (_____) _____-_____

All 1985 Back Issues are $10.00 each, when you order them separately. If you purchase all available 1985 issues as a set, the price is $50.00. All 1986 Back Issues are $7.00 each. All prices include postage and tax.

The following issues are currently available. Check those you wish to order:

( ) FEBRUARY 1985
( ) MARCH 1985
( ) APRIL 1985
( ) MAY 1985
( ) JUNE 1985
( ) JULY 1985
( ) AUGUST 1985
( ) SEPTEMBER 1985
( ) OCTOBER 1985
( ) NOVEMBER 1985
( ) DECEMBER 1985 (includes 1985 Annual Disk and 1985 Cumulative Index)
( ) JANUARY 1986
( ) FEBRUARY 1986
( ) MARCH 1986
( ) APRIL 1986
( ) MAY 1986
( ) JUNE 1986
( ) JULY 1986

$_____ TOTAL AMOUNT ENCLOSED

( ) VISA ( ) MasterCard # _____

Expiration Date _____ Signature _____

ALL ORDERS MUST BE PREPAID. MAKE CHECKS PAYABLE TO ASHTON-TATE (IN U.S. DOLLARS DRAWN ON U.S. BANK). PLEASE ALLOW SIX (6) WEEKS FOR DELIVERY OF YOUR ORDER.

NOTE: Information contained in ***TechNotes*** is current as of the issue in which it appears. Therefore, some information found in the Back Issues may no longer be valid (such as Update Policies, etc.).

CUMULATIVE INDEX

January 1986 through August 1986

Articles

Subject	Product	Issue	Page	Title and Description
@isabend	**Framework II**	Feb	38	**The Protean Powers of @isabend** *Explanation of this powerful function which tests for error conditions*
@panel1/@panel2	**Framework II**	Jan	14	**Overview of New Framework II Features** *Explanation of these new functions*
ASSEMBLY LANGUAGE	**dBASE II**	Feb	13	**Interfacing with the Machine** *How to use the assembly language interface with dBASE II*
CATALOG	**dBASE III PLUS**	Apr	26	**An Example of Catalog Management** *Programs for working with the Catalog*
CHANGE SUMMARY	**dBASE III PLUS**	Jul	11	**Change Summary—Part One** *Part One covers anomalies reported in TechNotes July - October 1985*
		Aug	12	**Change Summary—Part Two** *Part Two covers anomalies reported in November 1985 - January 1986*
COMPUSERVE FORUM	**Other**	Jun	11	**The Ashton-Tate CompuServe Forum** *To provide an alternate means for you to obtain support*
DATA	**dBASE III**	Feb	31	**Data Encryption Strategies** *Programs for data security in dBASE III*
	dBASE III PLUS	Feb	11	**Toggling Between Table and Form Views** *Program to toggle between table view and form view in dBASE III PLUS*
DATA LIBRARY	**dBASE III PLUS**	Jun	22	**Creating a Database File Library** *Saving database file structures in a custom database file*
DATA TRANSFER	**Framework II**	Jul	24	**Symphony to Framework II Translation** *Translating data files from Symphony*
DATE RANGE	**dBASE III PLUS**	Aug	30	**Searching for the First Date Within a Range** *A program to search for dates within specified ranges*
dBASE TOOLS FOR C	**dBASE III PLUS**	Aug	38	**Advanced Sorting Primitives for dBASE TOOLS FOR C** *Advanced searching and sorting using C from dBASE III PLUS*
	Other	Apr	9	**Arrays in dBASE TOOLS FOR C** *Interfacing with the C language from dBASE III PLUS*
		Jun	25	**Searching, Sorting in dBASE TOOLS for C** *Searching and sorting using C from dBASE III PLUS*
DOS	**Other**	May	20	**Batch Files—The Easiest Power User Tool** *How to use batch files to eliminate unnecessary keystrokes*
		Mar	29	**Generating Menus in PC/MS-DOS** *How to use batch files to create menus in DOS*
FORMAT FILES	**dBASE III PLUS**	Jul	9	**Converting a Format File to Form Report** *Converting a format file into a printable form*
FRAMEWORK	**Framework**	Jan	7	**Interview with Robert Carr** *Evolvement of spreadsheets in Framework*
		Feb	3	**TechNotes Interview with Robert Carr** *Part 2, Framework creator discusses possible plans for the future*
	Framework II	Feb	28	**Framework II for Framework 1.x Users** *Discussion of transfering Framework 1.x files to Framework II*
		Feb	24	**Library Macros and Abbreviations** *How to use, edit, and create macros in the Library*
HELP	**dBASE III PLUS**	Aug	19	**The HELP System Revealed** *Map of the dBASE III PLUS HELP system*
HISTORY	**dBASE III PLUS**	Feb	7	**Using History in dBASE III PLUS** *Using HISTORY as an algorithm-building tool*
IIF( )	**dBASE III PLUS**	Mar	8	**The IIF( ) Function** *Examination of this new dBASE III PLUS function*
IMPORTING	**dBASE III**	Aug	35	**Importing Delimited Files with No Delimiters** *A program to import delimited text files without comma delimiters*
INSTALLATION	**dBASE III PLUS**	Apr	20	**Operating PLUS in a 256K Environment** *Tips for operation in this environment*

Articles

Subject	Product	Issue	Page	Title and Description
LIBRARY	**Framework II**	Apr	34	**A Guided Tour of the Library Cabinet** *A tour of this new Framework II feature*
		Jan	32	**Creating String Functions** *Adding functions and macros to Framework II Library*
MAILING LABELS	**Framework II**	Mar	14	**Automating Mailing Labels** *FRED program to ease the generation of mailing labels*
MENUS	**dBASE III PLUS**	Mar	12	**SET COLOR from a Menu** *A program for defining color from a pull-down menu*
	Framework II	Jan	14	**Overview of New Framework II Features** *New options in the Disk, Create, and Edit pull-down menus*
		Feb	21	**Overview of New Framework II Features** *New options in the Locate, Frames, Graph, and Print pull-down menus*
MODIFY STRUCTURE	**dBASE II**	Jan	10	**An Alternative to MODIFY STRUCTURE** *A dBASE II program to emulate this command*
NOVELL NETWORK	**dBASE III PLUS ADMINISTRATOR**	Jun	4	**Installing dBASE III PLUS on the Novell** *Solutions to errors during installation to the novell network*
PFS:FILE	**Framework II**	Mar	36	**Converting PFS Files to Framework II** *Program to transfer the files*
PRINTER DRIVERS	**Framework II**	Aug	3	**Framework II Printer Drivers** *A comprehensive discussion on installing the correct printer driver*
PRINTING	**dBASE III**	Mar	21	**Printing Program Listings** *Program to print dBASE III program files*
	dBASE III PLUS	May	24	**Sending Screens to the Printer in dBASE** *Printing screens by LOADing and CALLing a DOS function*
	Framework II	Aug	10	**Common Printer Problems and Their Solutions** *Questions and answers about common printer problems*
PROTECT	**dBASE III PLUS**	Aug	23	**Protect Notes** *Notes on Protect for the advanced user*
RANDOM NUMBERS	**Other**	Jan	17	**Random Numbers Generation for dBASE** *Random number generation in dBASE II and dBASE III*
RECORD LOCKING	**dBASE III PLUS ADMINISTRATOR**	Aug	25	**Automatic Record Locking While EDITing** *A program that implements record locking automatically while EDITing*
REPORT FORM	**dBASE III**	May	14	**Techniques for Customized REPORT FORMs** *Using calculated fields and formatting techniques to produce a report*
REPORTING	**dBASE III PLUS**	Jul	3	**Custom Grouped Reports in dBASE III PLUS** *Defining custom reports for control break reporting*
		Apr	20	**Listing a Database File in a Verticle Form** *An alternative way to LIST a database file*
		Aug	27	**Multiple Across Reporting Using LABEL FORM** *Using the LABEL FORM generator to print reports*
RUNDBASE	**Framework II**	Mar	27	**Using Rundbase on a Hard Disk** *Program to run a hard-disk based dBASE III from Framework II*
SCREEN DRIVERS	**Framework II**	Jul	16	**Framework II Screen Drivers** *Installing the correct screen driver, with color, for your hardware*
SET COLOR	**dBASE III PLUS**	Mar	12	**SET COLOR from a Menu** *A program which generates a menu for color definition*
SPREADSHEET	**Framework II**	Feb	34	**Data Tables in Framework II** *Creating data tables for "what-if" modeling*
		Mar	25	**Exchanging Numbers and Text in a Cell** *Useful macros for Framework II spreadsheets*
TABS	**Framework II**	Jan	34	**Removing Tabs from a Word Frame** *Program used on text files before or after transmitting via Telecomm*
		Mar	18	**Variable Tab Stops** *Creating variable tab stops in Framework II*
TELECOMMUNICATIONS	**Framework II**	Jan	34	**Converting PFS files to Framework II** *Program used on text files before of after transmitting*
		Apr	3	**Telecommunications Primer** *An introduction to hardware and software setup*

Articles

Subject	Product	Issue	Page	Title and Description
TELECOMMUNICATIONS	**Framework II**	May	3	**Telecommunications Primer—Part Two** *Using Framework II telecommunications to upload/download data*
		Jun	16	**Telecommunications Primer-Part Three** *Defining logon macros to automate your telecommunications tasks*
VIEWS	**dBASE III PLUS**	Mar	4	**Understanding Views** *Discussion of views as they enhance relational operations*
WORD FRAMES	**Framework II**	Apr	39	**How to Use Boxes in Word Frames** *A creative way to work with documents*

Anomalies

Product	Entry	Issue	Page	Description
dBASE II	APPEND FROM	Jan	45	*Cautions regarding APPENDing numeric data in v. 2.43**
	COUNT	Feb	43	*In DOS ver. 3.x, COUNT may take longer to execute the second time*
	ERROR MESSAGE	Feb	42	*" 'TO'PHRASE NOT FOUND": Memory variable corruption*
	MEMORY VARIABLES	Feb	42	*Corruption of numeric memory variables in dBASE II, version 2.43**
	RENAME.COM	Feb	43	*This file is not on the RunTime disk, as the manual states*
	SKIP	Apr	47	*Cannot have a space preceding the argument of the SKIP function*
dBASE III	BROWSE	Jan	45	*Problems using the SEEK option with SET DELETED ON*
	ERROR MESSAGE	Jan	46	*"Record is out of range": trying to GOTO a very large number*
	FIND	Jan	45	*With numeric INDEX, FIND <cstring> will FIND first zero valued record*
	GOTO	Jan	46	*Attempting to GOTO a very large number may lock the computer*
	INDEX	Jan	45	*With numeric INDEX, FIND <cstring> will FIND first zero valued record*
	INSTALL	Jan	46	*Developer's Release: INSTALL does not copy DBRUN.OVL*
	PARAMETERS	Jan	46	*Cannot wrap PARAMETERS line with semi-colon*
	RUNTIME+	Jan	46	*DBRUN.OVL is not copied during Developer's Release INSTALL*
	SEEK	Jan	45	*Problems using the SEEK option in BROWSE with SET DELETED ON*
	SET DELETED ON	Jan	45	*Problems using the SEEK option in BROWSE with SET DELETED ON*
dBASE III PLUS	@...SAY/GET	Apr	48	*Should not use line 0 with SET SCOREBOARD ON*
	CREATE/MODIFY SCREEN	Apr	48	*The <new field> option creates a one-character field*
	CREATE/MODIFY VIEW	Apr	48	*Can only include files from the current directory. Cannot delete a FILTER in MODIFY VIEW*
	ERROR MESSAGE	Apr	48	*"Alias already in use": Accessing another subdirectory with views*
		Apr	48	*"Record not in index": Using SET CENTURY ON with DTOC( ) to index file*
	FOUND( )	Apr	48	*Will return .F. if first record is deleted and SET DELETED is on*
	INDEX	Apr	48	*Cannot use SET CENTURY ON with DTOC( ) to create an index file*
	READKEY( )	Apr	49	*Returns 270 if operation is terminated by Ctrl-End or Ctrl-W*
	SET HELP	May	33	*The full-screen SET command HELP screen is displayed on SET errors*
	SET SCOREBOARD	Apr	47	*Should not address line 0 with SET SCOREBOARD ON*
Framework II	.TCM SETUP FILES	May	34	*Telecommunication setup files with imbedded spaces cause DOS errors*
	@drawgraph	Jun	42	*Multiple lines in a graph title will display graphic characters*
	@run	Jun	42	*@run while Recalc Formula During Printing is ON causes screen flicker*
	@setformula	May	33	*A Return after the comma may hide the formula from view*
	@trace	May	34	*@int and @abs may return incorrect results if @trace is ON*
	DICTIONARY	Jun	42	*An invalid filename for the Personal Dictionary will cause an error*
	EDIT:UNDO	Jul	52	*UNDOing and EDIT FORMULA (F2) accidentally can hang the computer*
	ERROR MESSAGE	Jun	42	*"Personal Dictionary not found. Create a new one and continue? (y/n)"*
	EXTENDED MEMORY	Jul	52	*Problems printing spreadsheets with extended memory and cell overlap*

Letters

Product	Entry	Issue	Page	Description
dBASE II	DATA CORRUPTION	Jun	47	*Corruption of data may occur using any version on a network*
	DATA DISPLAY	Feb	44	*Old data displaying in a file with zero records*
	INDEX	Jun	48	*INDEXing a character field with a descending numeric field*
	NUMERIC ACCURACY	Jan	49	*Versions prior to 2.43* have 10 digits of numeric accuracy*
	PRINTING	Apr	56	*Sending line feeds (without carriage returns) to the printer*
	REPORT FORM	Jan	48	*Incorrect totals in versions prior to 2.43**
	SED	Mar	44	*Work-around for hard tabs created in SED*
	WANG PC	May	41	*Redefining WANG PC function keys using version 2.41*
dBASE III	CHARACTER STRINGS	Mar	45	*Removing preceding blanks from character strings*
	CHR( )	Jun	46	*Sending a CHR(12) to a printer resets PROW( ) to zero*
	DATA CORRUPTION	Jun	47	*Corruption of data may occur using version 1.1 on a network*
	ERROR MESSAGE	Jan	48	*"NO FIND": Character index key length must be a multiple of 4*
		Jan	48	*"Internal error, illegal opcode": Char INDEX must be a multiple of 4*
	FUNCTION KEYS	Jun	48	*Using the semicolon to program the function keys*
	IMPORT/EXPORT	Feb	44	*Transferring data from Lotus 1-2-3 to dBASE III*
	INDEX	Jan	47	*Character INDEX key length must be a multiple of 4*
		Apr	56	*Creating index files in a subdirectory*
		Jun	48	*Indexing a character field with a descending numeric field*
	INSTALL	Jan	48	*Problems with Tallgrass: set Duplicate Directories OFF*
	LABEL FORM	Mar	45	*Using two database files with LABEL FORM*
	SET	Jan	49	*Accessing SET parameters from within program control*
	SET PATH TO	Apr	56	*Cannot use SET PATH TO to create index files in subdirectories*
	TRIM( )	May	40	*Right-justifying character fields with the TRIM( ) function*
dBASE III PLUS	@...SAY...GET	Jul	66	*Forced-choice response into a memory variable*
	DEBIT/CREDIT	Aug	66	*Modifying DBASE.MSG with DEBUG to modify the display of credits and debits*
	IIF( )	Apr	57	*Using IIF( ) for conditional indexing*
		Jul	66	*Using IIF( ) as an index key*
	INDEX	May	41	*Indexing on a date and time field*
	LTRIM( )	Mar	45	*This function removes preceding blanks from character strings*
	RELATION	Jul	67	*Relating multiple database files together to access over 128 fields*
	REPLACE	Aug	66	*Using AT( ) and STUFF() to REPLACE the letter "l" with the digit "1"*
	REPORTING	Aug	67	*Displaying totals in REPORT FORM with character output*
Framework II	@date1	May	40	*Bolding the system date in a header using @hl and Ctrl-W*
	@hl	May	40	*Bolding the system date in a header using @date1 and Ctrl-W*
	ASSEMBLY LANGUAGE	Mar	46	*Using Framework II word processor to write assembly language programs*
	IMPORT	Apr	55	*How to import Lotus 1-2-3 Release 2 files*
	MEMORY-RESIDENT	Feb	45	*May have problems using memory-resident programs with Framework*
		Jul	47	*Running memory resident utilities such as Sidekick or TimeFrame*
	NOVELL NETWORK	Jun	46	*Unable to correctly find the network disk drives on a Novell Network*
	TEXT FRAME	Mar	68	*A containing frame can hold more than one 64K text frame*
	SOFTWARE SUPPORT	Aug	65	*Explanation of long wait time on the phone calling Software Support*

On-Line Services

Product	Entry	Issue	Page	Description
dBASE II	RESIDENT EDITORS	Jun	52	*Using memory resident text editors to edit programs may cause errors*
	SET FORMAT	May	12	*Transfering files from Perfect Filer*
dBASE III	DUPLICATE ENTRIES	Jun	53	*Checking for duplicate entries during data entry*
	EOF( )	Jun	54	*Encountering an end-of-file error after a SKIP*
	ERROR MESSAGE	Jun	54	*"End of file encountered."*
	MEMO FIELDS	Jun	55	*Creating a new memo file after loosing the original .DBT file*
	PRINTING	Jun	54	*Setting the configuration of the printer correctly before printing*

On-Line Services

Product	Entry	Issue	Page	Description
dBASE III PLUS	CATALOG	Jul	57	*Limitations of the CATALOG feature*
	CONFIG.DB	Jun	55	*Modifying CONFIG.DB to suppress ASSIST from loading*
	COPY PROTECTION	Jun	52	*Install and uninstall using the same System Disk #1*
	DCONVERT.EXE	Aug	56	*Converting from dBASE II to dBASE III PLUS*
	FUNCTIONS	Aug	57	*Explanation and example uses of the @R function*
	GRAPHIC BOARDS	Jun	55	*Using the SIGMA 400 graphics board will hang the computer*
	INDEX	May	13	*The index key length problem is corrected in PLUS*
	LABELS	Jul	56	*Using the IIF( ) function to print out labels*
	MEMO FIELDS	Jul	58	*Using ? to display memo field to an ALTERNATE file to strip out spaces*
	NETWORK	May	12	*Network commands are ignored in the Sampler*
	NUMERIC VARIABLES	May	12	*Converting a numeric field to a character field for INDEXing*
	REPORT FORM	Jul	57	*Using the semicolon in a REPORT FORM to return to a new line*
	SEEK	Jul	57	*Indexing with UPPER( ) to make SEEKing easier*
	SET FILTER TO	May	11	*Using SET FILTER TO and SET RELATION TO together*
	SET ODOMETER TO	Jun	54	*SETting the ODOMETER to display percentages*
	SET RELATION TO	May	11	*Using SET FILTER TO and SET RELATION TO together*
	TIME( )	Jun	52	*Using the TIME( ) function in calculations*
	TRANSFORM( )	May	13	*Using the TRANSFORM( ) function to print reports*
dBASE III PLUS ADMINISTRATOR	LOCKING	Jun	54	*Record locking is not implicit in the Administrator*
Framework II	@scan	May	9	*Using the @scan formula to filter a database*
	@setselection	Aug	59	*This function does accept variable names as a parameter*
	CONTROL CHARACTERS	May	8	*Filtering timing characters during telecommunication*
	CONTROL CODES	Jun	51	*Using the ALT key with the numeric keypad to embed control codes*
	DICTIONARY	Jul	54	*Additions to the personal dictionary are automatic*
	EXTENDED MEMORY	Jul	55	*Using the RamPage and DESQview with extended memory*
	IBM COLOR DISPLAY	Jun	50	*Color will display only if the correct screen driver is installed*
	IBM COLOR GRAPHICS	Jun	50	*Can display foreground color only using the IBM CGA*
	MAILMERGE	Jul	56	*Using a database in conjunction with user input for billing*
	MOUSE	May	10	*Using a mouse*
	NOVELL NETWORK	Jul	53	*Using the MAP command to assign network drives to Framework II*
	PRINTING	Jul	55	*Printing spreadsheets using Sideline*
		Aug	57	*Setting switches on the Prowriter 8510 to suppress extra line spacing*
	SCREEN DRIVERS	May	10	*Setting up the correct screen driver for the IBM EGA*
	TELECOMMUNI-CATIONS	Jun	51	*Waiting for trap characters in Telecommunications*
	TERMINAL EMULATION	May	9	*Using terminal emulation during telecommunication*
	TIMEFRAME	Aug	58	*Information on upgrading TimeFrame for Framework II*
Other	COMMAND.COM	Jun	51	*COMMAND.COM must be in the root directory for the computer to boot*
	DOS	Aug	55	*A known DOS problem surfaces when using the RUN command*

Sample Programs

Product	Entry	Issue	Page	Description
dBASE II	CSEED2.PRG	Jan	17	*Program to create a seed value for random number generation*
	EVAL2.PRG	Jan	22	*Calculates the standard deviation and mean value of random numbers*
	HEC-DEC.CMD/PRG	Feb	18	*Converts hex program into a dBASE II POKE sequence*
	LEADZERO.PRG	Feb	20	*Program to replace leading blanks with zeros*
	MODSTUC.PRG	Jan	10	*Program to emulate MODIFY STRUCTURE*
	RANDOM2.PRG	Jan	18	*Program to generate a random number*
	WINNERS.PRG	Jan	20	*Program to select random winners from a database file*
dBASE III	BUBSORT.PRG	Jun	30	*Performs a bubble sort on an array using dBASE TOOLS FOR C*
	CBINARY.PRG	Jun	27	*Uses dBASE TOOLS FOR C to implement a binary search using arrays*
	COMMAS.PRG	Aug	36	*Imports an undelimited, comma separated text file into dBASE III*
	COMPARE.PRG	Jun	28	*Uses dBASE TOOLS FOR C to compare two adjacent array elements*
	CSEED3.PRG	Jan	19	*Program to create a seed value for random number generation*
	DECRYPT.PRG	Feb	33	*Data decryption program for dBASE III*

Sample Programs

Product	Entry	Issue	Page	Description
dBASE III	ENCRYPT.PRG	Feb	31	*Data encryption program for dBASE III*
	EVAL3.PRG	Jan	23	*Calculates the standard deviation and mean value of random numbers*
	EXCHANGE.PRG	Jun	29	*Uses dBASE TOOLS FOR C to exchange the values of two array elements*
	FILEPROC.PRG	Mar	22	*Procedure file prints program listings*
	PRGPRINT.PRG	Mar	21	*Program to display menu selection for print program listings*
	RANDOM3.PRG	Jan	20	*Program to generate a random number*
dBASE III PLUS	COLOR.PRG	Mar	12	*Program to SET COLOR from a menu in dBASE III PLUS*
	CPDELIM.PRG	May	28	*COPY a comma separated text file without delimiters*
	DATABASE.PRG	Apr	28	*Maintain files in the Catalog management application*
	DATEFIND.PRG	Aug	32	*Moves the record pointer to the first date within a specified range*
	DISPLAY.PRG	Apr	30	*Program to manipulate data in the Catalog*
	ERR.PRG	Apr	32	*Program to handle errors in the Catalog management application*
	FREEARRAY.PRG	Apr	16	*Procedure to remove an array from memory*
	LIST.PRG	May	25	*Lists a screen and prompts for a screen dump to the printer*
	LISTFORM.PRG	Apr	23	*Program to list a database file in a vertical form*
	LISTHIST.PRG	Feb	8	*Program to LIST HISTORY to a file in dBASE III PLUS*
	LOADARRAY.PRG	Apr	14	*Procedure to load an array from a database file*
	MAIN.PRG	Apr	27	*Main driver program for the Catalog management application*
		Aug	31	*Accepts the starting and ending dates for Datefind.PRG*
	MAKEARRAY.PRG	Apr	12	*Procedure to create a new array*
	MULTIRPT.PRG	Aug	28	*Prints multiple records in columns across the page*
	NEWPAGE.PRG	Apr	25	*Program to print page top called from Listform.PRG*
	OPTION.PRG	Apr	29	*Program to allow access to index files and queries in the Catalog*
	PAGETOP.PRG	Jul	8	*Prints the page and column headings for Sales Report.*
	REMOVE.PRG	Jun	34	*Removes any set of characters from within a target string*
	SALES.PRG	Jul	7	*Prints a two-level grouped report*
	SAVESTRU.PRG	Jun	23	*Creates a library of database file structures of the current directory*
	SEEKARRAY.PRG	Apr	15	*Procedure to find an element within a defined array*
	VIEW.PRG	Feb	12	*Program to toggle between table view and form view*
	WRAP.PRG	Apr	24	*Program to wrap long strings, called from Listform.PRG*
dBASE III PLUS ADMINISTRATOR	AUTOERR.PRG	Aug	26	*Error routine for Autolock.PRG*
	AUTOLOCK.PRG	Aug	25	*Provides a facility for automatic record locking while EDITing*
Framework II	ALT-B (BOX MACRO)	Apr	39	*Macro to draw boxes in word frames*
	CALCSIZE	Jul	47	*A macro to calculate the overall width of a spreadsheet*
	COPYPROG	Aug	53	*A macro to backup files from hard disk to floppy*
	DATA TABLE	Feb	35	*Program to create data tables in Framework II*
	ELAPSED_TIME	Jul	46	*A macro to determine elapsed time between a start and end date or time*
	ENGLDATE	Aug	52	*A formula to compare English dates*
	HEX	Apr	37	*FRED function to convert a decimal number to hex*
	LABELPGM	Mar	14	*Program to automate mailing labels*
	LSTRING	Jan	25	*Function to extract the left portion of a character string*
	MID STRING	Jan	31	*Macro to extract the mid portion of a string in word frames*
	NAMED RANGES	Jul	45	*Creates a named range to be used in Framework II programs*
	PFS TRANSFER	Mar	37	*Program to transfer files from pfs:file to Framework II*
	RSTRING	Jan	26	*Function to extract the right portion of a character string*
	RUNDBASE	Mar	27	*Running dBASE from within Framework II*
	STRING FUNCTIONS	Jan	24	*User-defined string functions used in spreadsheet and database frames*
	SYM_TO_FW	Jul	26	*Symphony to Framework II translation program*
	TABSTOP	Mar	18	*Macro for putting variable tab stops into word frames*
	TABSTRIP	Jan	33	*Program to remove tabs from a word frame*
	WORD CHECK	Jul	48	*A macro to check the spelling of a word or phrase*
Other	CHOOSE.COM	Mar	33	*An assembly language routine used to choose menu options with MENU.BAT*
	MENU.BAT	Mar	32	*DOS batch file is controlling program for front-end menu system*
	MENU.DAT	Mar	33	*DOS batch file, called from MENU.BAT, displays menu screen*

Usage Tips

Product	Entry	Issue	Page	Description
dBASE II	@...SAY	Jan	37	*Patches to access line 24 with @...SAY in versions 2.4 or higher*
		Jan	37	*"@ row,column SAY ERASE" only clears one line in CP/M-80 formats*
	NUMERIC VARIABLES	Mar	40	*How to left-justify numeric variables*
	SUBDIRECTORIES	Mar	39	*Using logical drive designators as subdirectories with DOS 3.1*
dBASE III	ASSIST	Mar	40	*Commands executed in ASSIST can be recorded in HISTORY*
	COPY	Jan	38	*Method to COPY fields from one database file to another*
	DELETE	Jan	39	*Deleting records based on values in a related database file*
	SET PATH TO	Jan	40	*If SAFETY is ON, some commands will search the path before writing to disk*
Developer's Release	DBRUN.COM	Jan	40	*This file, not DBRUN.EXE, should be renamed as per the documentation*
	DBRUN.EXE	Jan	40	*This hidden file will not show up with DIR and cannot be renamed*
	READ	Jan	40	*In Developer's Release, a READ must have active GETs*
		Jan	40	*Cannot issue a READ from within HISTORY in Developer's Release*
	REINDEX	Jan	40	*Must reissue the SET ORDER TO command if you REINDEX*
	SET ORDER TO	Jan	40	*Use REINDEX to SET ORDER TO its original state*
dBASE III PLUS	43-Line Mode on EGA	May	27	*Toggling the IBM Enhanced Graphics Adapter's 43-line display mode*
	@...SAY/GET	May	27	*Large SCOREBOARD area conflicts with @...SAY/GET to line 0*
	ACCESS DISK	Apr	42	*Disk manufacturing error affected products with certain serial numbers*
	APPEND FROM	May	27	*Files without delimiters will be correctly APPENDed*
	ASSIST	Apr	43	*Building search conditions with date fields*
		Apr	43	*Cannot build a search condition for blank spaces*
	CASE	Apr	43	*Omission of DO CASE or ENDCASE in a CASE structure affects commands*
	CHARACTER STRINGS	Jun	33	*Removing characters from a field or memory variable*
	CONFIG.DB	Apr	44	*Using the keyword COMMAND in CONFIG.DB will alter the initial sign-on*
	CONTROL CODES	Apr	42	*Control codes to the printer with @...SAYs are counted as character*
	COPY STRUCTURE	May	28	*Recreating a file structure with COPY STRUCTURE EXTENDED*
	COPY TO [DELIMITED]	May	28	*COPY a comma separated text file without delimiters*
	CREATE FROM	May	28	*CREATE a new file from another containing the structure as records*
	CREATE SCREEN	Jun	32	*Using double quotation marks will cause an error*
	CTOD( )	Apr	43	*Non-date entries are accepted as arguments in ASSIST*
	DO WHILE	Apr	44	*DO WHILE EOF( )=.T. returns "Invalid Operator"*
	ERROR MESSAGE	Apr	44	*"Invalid Operator": Results from improper syntax of DO WHILE statement*
		Jun	32	*"Unrecognized phrase/keyword in command": results from using (")*
		Jun	32	*"Variable not found": results from using "-" in a filename*
		Jun	33	*"Invalid operation on a memo field": results from executing JOIN*
		Jun	33	*"Memo file can not be opened": returned when .DBT file is not found*
	FILENAMES	Jun	32	*Database filenames with an embedded dash cannot be used*
		Jul	41	Accessing filenames with no extensions
	INKEY( )	Aug	50	*This function seems to return an incorrect value after executing a WAIT*
	INPUT	Apr	44	*INPUT does not accept a Return as valid input*
	INTERACTIVE MODE	Apr	44	*Cannot continue dot prompt commands with a semi-colon*
	JOIN	Jun	33	*Specifying a memo field in the FIELDS clause returns an error*
	MODIFY VIEW	Apr	46	*Will return error message if any components of the view have changed*
	PUBLIC	May	30	*Variables SAVEd as PUBLIC will be RESTOREd as PUBLIC*
	REPLACE	May	31	*Using REPLACE to update a many-to-one related database file*
		Jul	42	*Using REPLACE to change field contents in other work areas*
	REPORT FORM	Jun	34	*Changes made to the REPORT FORM*
	RESTORE FROM	May	30	*Variables remain PUBLIC once RESTOREd if declared so before a SAVE*
	SAVE TO	May	30	*Variables SAVEd as PUBLIC will be RESTOREd as PUBLIC*
	SET COLOR	May	30	*A table of the default color attributes shows changes from dBASE III*

Usage Tips

Product	Entry	Issue	Page	Description
dBASE III PLUS	SET DATE FRENCH	Apr	45	*This command is now the same as SET DATE BRITISH*
	SET FIELDS TO	Apr	45	*SET FIELDS TO <field list> automatically SETs FIELDS ON*
	SET ODOMETER TO	Jun	35	*Setting the step count of the percentage display*
	SET ORDER TO	Apr	45	*Must reposition the record pointer after this command*
	SET RELATION TO	May	31	*A syntax error is returned if SET RELATION is used without TO*
		Aug	48	*SET RELATION TO relates two database files in three different ways*
	SET SCOREBOARD	Apr	45	*Clears the entire screen*
		May	27	*Using @...SAY to line 0 will not write in the SCOREBOARD area*
	SET STATUS	Apr	45	*Clears the entire screen*
	SET VIEW TO	Apr	46	*Cannot make changes to any component parts of a view*
dBASE III PLUS ADMINISTRATOR	BACKUP AND RESTORE	Jul	43	*Using BACKUP and RESTORE on disks that are not copy-protected*
	DEBUGGING	Jul	43	*Debugging commands to assist you during installation problems*
	DOCUMENTATION	Jul	43	*Corrections to reported documentation errors*
	ERROR MESSAGE	Jun	36	*"Network is busy": indicates no more tasks may be accepted by server*
		Jun	36	*"Record is in use by others": returned if RLOCK( ) was used by another*
		Jun	38	*"Syntax Error": generated by inappropriately using SET PRINTER TO*
		Jul	42	*"File is in use by another": Attempting to USE a read-only file*
		Jul	44	*"Record in use by another": Results when LISTing a locked record*
		Jul	44	*"NET 803: Network path not found": MODIFY STRUCTURE on IBM PC-NET*
		Jul	44	*"File is already open": Occurs using MODIFY STRUCTURE on IBM PC-NET*
		Jul	45	*"Record is in use by another": Using FIND with SET DELETED ON*
	GOTO	Jun	36	*Use of locked records by other users with RLOCK( ) will return an error*
	LIST	Jul	44	*LISTing a locked record will display "Record in use by another"*
	MODIFY COMMAND	Jun	37	*Attempting to modify a read-only file will yield inconsistent results*
	MODIFY STRUCTURE	Jul	44	*Generates the error "NET 803: Network path not found" on IBM PC-NET*
	NETWORKS	Jun	36	*A busy network file server may become unreliable*
	PROTECT	Jun	37	*The password entry screen is bypassed if DBSYSTEM.DB is not found*
	RLOCK( )	Jun	36	*A record locked by another user with RLOCK( ) will return an error*
	RUN	Jun	37	*The Novell system closes all open files unless EOJOFF is executed*
	SET DELETED	Jul	45	*SET DELETED ON with duplicate records can affect the FIND command*
	SET PRINTER TO	Jun	38	*The command SET PRINTER TO \\SPOOLER is appropriate only to Novell*
Framework II	@date	Aug	52	*Comparing English date strings*
	@if	Jun	38	*Using @select to maintain readability of cascading @if structures*
	@performkeys	Jun	39	*Displaying the keyname of the last key pressed*
	@select	Jun	38	*Using @select to maintain readability of cascading @if structures*
	@sum	Mar	42	*Cannot remove the last column/row referenced by @sum*
	@time	Jan	42	*Displaying times with the Numbers pull-down menu*
	@trace	Jul	50	*A macro to turn TRACE on and off*
	@writetextfile	Jun	40	*The correct way to use variables with @writetextfile*
	CONDENSED PRINT	May	32	*Condensed Print takes precedence over Near Letter Quality on LQ1500*
	COPYING	Jun	40	*Copying the source information of multiple columns is not allowed*
		Aug	53	*A macro to backup files from the hard disk to floppy disks*
	DELETING SPACES	Jul	49	*Using hard spaces to delete spaces as words*
	DICTIONARY	Mar	42	*May take a long time to save the personal dictionary the first time*
	DOS ACCESS	Jun	41	*The DOS Access screen will clear if ANSI.SYS is used as a device*
	ELAPSED TIME	Jul	46	*A function to calculate elapsed time accurately*
	ERROR MESSAGE	Jun	40	*"Frame too large to be printed": results from sizing a graph to large*
	EXTENDED MEMORY	Mar	42	*Memory requirements for extended memory*
	GRAPHING	Jun	40	*Unexpected limitations printing graphs when using an EPSON LQ1500*
	IBM 3270t	Jan	44	*This screen driver is found under "All Other IBM Drivers" in SETUP*

Usage Tips

Product	Entry	Issue	Page	Description
Framework II	IMPORT	Mar	41	*Importing Framework 1.x files to Framework II*
	IMPORT/EXPORT	May	32	*Procedure to correctly transfer Lotus files*
	MAILMERGE	Mar	43	*Some Print Output Options have no effect on Mailmerge or Print Labels*
	MAILMERGE	Jul	49	*Right-justifying numbers for use with a mailmerge*
	PAR FILES	Jan	44	*How to read .PAR files from Framework into Framework II*
	PRINTING	Aug	54	*A macro for printers with sheet feeders*
	PROTECT FROM EDITING	May	32	*Empty spreadsheet cells cannot be protected until they contain data*
	RANGES	Jul	45	*Creating a Library macro to use named ranges in FRED programs*
	RUNNING DBASE	Jun	41	*Modifying RUNDBASE.FW2 to allow dBASE III to be run from a hard disk*
	SETUP	Jan	42	*Selecting screen color parameters in SETUP*
	SIZING COLUMNS	Mar	42	*Changing several spreadsheet or database columns at once*
	SPELL CHECKER	Jul	48	*A function to spell check a word without interrupting your work*
	SPREADSHEETS	Mar	42	*Copying fractions from text into spreadsheets*
		Jul	47	*A function to calculate the overall width of a spreadsheet*
	SUB/SUPERSCRIPTS	Mar	41	*How to print subscripts and superscripts from Framework II*
		Aug	51	*A macro that generates the proper codes for superscripts and subscripts*
	TELECOMMUNI-CATIONS	Mar	43	*Converting .PAR files to .TCM files*
		Jul	48	*Setting the Telecomm parameters for Point and Dial*
	VIRTUAL MEMORY	Mar	41	*Cannot use virtual memory to import Framework 1.x files*

Miscellaneous

Subject	Issue	Page	Description
CORRIGENDA	Jan	36	Corrections to September, November, and December 1985 issues
	Feb	27	Corrections to the March 1985 issue
	Mar	3	Corrections to the July 1985, October 1985, and December 1985 issues
	Apr	53	Corrections to the December 1985 issue
	May	7	Corrections to the March 1986 issue
	Aug	54	Correction to the June 1986 issue
INTERNATIONAL	Jul	64	International Support - List of international subsidiary offices
	Aug	63	Reassigning keyboard characters in Framework
	Aug	64	Information about Ashton-Tate in Asia
MULTIMATE	Apr	50	Features of MultiMate Advantage update (version 3.60)
	Apr	52	Where to look when you have technical difficulties
	Apr	52	About MultiMate Technical Support—Supported products, when to call
	Apr	53	Customer Service versus Technical Support
	May	36	Listing and description of communications boards from MultiMate
	Jun	43	MultiMate version 3.30 printer of the month - HP Thinkjet
	Jun	43	MultiMate printers of the month (HP)
	Jun	44	MultiMate version 3.30 printer of the month - HP Laserjet
	Jul	59	MultiMate Magic - Technical tidbits in response to "WHAT IF" situation
	Jul	62	MultiMate PC Questions - Commonly asked questions about MultiMate
	Aug	59	Common Printer Problems - Questions and answers about printers
	Aug	62	MultiMate Magic - Technical tidbits in response to "WHAT IF" situations
NEW PRODUCTS	May	37	Expanded products line from Multimate for the IBM PC family
ON-LINE SUPPORT	Jan	3	Information pertaining to CompuServe, The Source, and Onepoint
SUBSIDIARY OFFICES	Jan	56	List of Ashton-Tate international subsidiary offices
SUPERLoK	Jul	28	Technical information about SUPERLoK, the copy-protection scheme
TRAINING CENTERS	Jan	54	Listing of Ashton-Tate Authorized Training Centers
	Mar	47	Listing of Ashton-Tate Authorized Training Centers

Notes

Notes

TechNotes
P.O. Box 3729
Escondido, CA 92025-0929

ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 27
CULVER CITY, CA